The Lean Distributed Scrum (LDScrum) Model: Investigating the Relationship between Lean and Agile Software Development in a Distributed Development Context

Michael T. Lane
B.Sc., MMII Grad Dip., MSc., PMP

Supervisor: Professor Brian Fitzgerald
Co-supervisor: Professor Pär Ågerfalk

Department of Computer Science and Information Systems

Faculty of Science and Engineering

UNIVERSITY of LIMERICK
OLLSCOIL LUIMNIGH
Abstract

Agile software development (ASD) has emerged as a practice-led initiative which offers great promise in improving software productivity. However some confusion exists as to its relationship with Lean Software Development (LSD). Some treat LSD as more or less synonymous with ASD whereas others view LSD as a different concept. The definition and positioning of LSD relative to ASD is important as it gets to the heart of software development as craft versus science debate. The purpose of this thesis is to present a deeper understanding on the relationship between these two paradigms.

It is proposed that LSD is more management philosophy than method. In order to facilitate this exploration, a set of core LSD values is induced from literature and proposed to ‘define’ LSD, much as the agile manifesto values unified and defined so-called ‘lightweight methods’ for ASD. In order to narrow the focus of this investigation, the broad management philosophy of LSD is explored in the context of a very specific instance of ASD: the application of the most dominant ASD method (Scrum) in the management of projects that use globally distributed teams (GSD).

The main deliverables from this interpretive study are a research framework and a thick description of empirical findings in this regard. The Scrum method is described as comprising of a set of core Scrum practices. The research framework leverages method rationale analysis theory in order to propose LSD values that may underpin goals pursued by the application of those Scrum practices within globally distributed teams. It is recommended that this research framework be applied to further investigations into the influence of LSD on other Scrum manifestations as well as other ASD methods.

An interpretive case study approach was applied to examine different Scrum teams and projects. Although all LSD values were found to be present in the case, certain values were more prevalent than others. ‘Effective collaboration’ (reduction of distance) and ‘Person-focus’ (individual and team empowerment) were the most prevalent LSD values observed. ‘Continuous improvement’ was also very evident. One project within the case showed very little awareness of the core LSD values of ‘Flow of value’ and ‘Waste reduction’. Across the entire case, the value of ‘Data-driven decisions’ was not seen to have a strong influence on developer intentions. This lack of statistical rigour may reflect the craft versus science controversy in relation to software development.
Dedication

‘Do not disturb - Angry Author Inside!’
(Sign stuck on my office door by the kids)

In order to tackle this study effectively, it very quickly became clear that total immersion was the only way to conduct the work. Such an intrusion on our family life could not and would not have been acceptable without the love, leadership, whole-hearted generosity and positive support provided by my wife, Noreen. I dedicate this thesis to you, Noreen and to my children, Michael, Ellen and Sarah for embracing the concept of ‘student Dad’ and making the pursuit of this PhD become a normal part of our everyday family life.
Acknowledgements

The risk of omission is always a concern when delivering statements of appreciation. Let me apologize in advance for any such blunders and by way of coverage, firstly thank everybody that has offered me encouragement, advice and support throughout this project. I feel very privileged to have been surrounded by so many positive individuals within the Irish and European research communities, work colleagues past and present and friends from my own community who expressed interest in my studies.

A major portion of the learning that I have achieved is as a result of the excellent supervision provided by Professor Brian Fitzgerald and Professor Par Agerfalk. Their intelligent non-intrusive guidance ensured that this study was an evolution of thoughts and ideas that encouraged me to consider my work carefully and appreciate the importance of research craftsmanship.

I would like to thank my examiners, Professor Bjorn Lundell and Dr. Patrick Buckley for valuable feedback arising from their diligent and thorough review of my work.

Of the many people that supported me in this endeavour, two individuals deserve special mention:

- Ms. Annette Mc.Elligott (Head of the CSIS department at UL) for her motivation and encouragement in my pursuit of this PhD.
- Mr. Paul Newton (Manager at I-E-S) for his support and coordination of many of the empirical research activities.

My parents, Michael and Jane have been an enormous influence on all of my educational pursuits. Their lifelong promotion of education and their wisdom in building confidence in all their children from an early age is undoubtedly a major reason that I set out on this journey.

Finally, to my wife Noreen and my children, Michael, Ellen and Sarah. I cannot thank you enough for giving me the space and time to pursue this work. It may be my name on the cover but I am in no doubt that all of the family played a part in making this thesis possible.
Declaration

This thesis represents the author’s own original work. Where this work has built from the scholarship of others, these sources are acknowledged.
# Table of Contents

## CHAPTER 1 - INTRODUCTION

1. 1 Motivations for study .............................................................. 1

### 1.1.1 Philosophical foundations of LSD and ASD ........................... 1

### 1.1.2 Application of LT and AT to manufacturing: LM and AM ........... 4

### 1.1.3 Application of LT and AT to Supply Chain Management (using LM and AM) ................................................ 5

### 1.1.4 Application of LT and AT to Software development (LSD & ASD) ............................................................... 6

### 1.1.5 Inconsistencies between LT and AT application ....................... 8

### 1.1.6 ASD method application used to explore LSD relationship .......... 10

1. 2 Overview of Areas Under Research ............................................. 13

### 1.2.1 Lean Software Development .................................................. 14

### 1.2.2 Agile Software Development .................................................. 17

### 1.2.3 Project Management of Agile methods .................................... 21

### 1.2.4 Global Software Development ............................................... 26

### 1.2.5 Background - summary ....................................................... 27

1. 3 Objective and Research Questions ................................................. 28

### 1.3.1 Research objective and key contributions ................................. 28

### 1.3.2 Research Approach and questions ......................................... 29

### 1.3.3 Study Work Breakdown Structure .......................................... 32

1. 4 Thesis Structure ............................................................................ 34

## CHAPTER 2 - LEAN SOFTWARE DEVELOPMENT

2. 1 Introduction .................................................................................. 36

### 2.2.1 Influence of operations management on LSD ......................... 37

### 2.2.2 Influence of Product Development on LSD ............................ 43

### 2.2.3 Influence of Project Management on LSD ............................. 45

### 2.2.4 Influence of Software Development on LSD .......................... 48

2. 3 Terminology .................................................................................. 54

2. 4 Principles and Values of LSD ....................................................... 58

2. 5 Summary ...................................................................................... 61

## CHAPTER 3 - SCRUM PRACTICES

3. 1 Introduction .................................................................................. 62

### 3.2.1 The origins of Scrum - New Product Development (NPD) characteristics ....................................................... 64

### 3.2.2 Agile software development (ASD) principles .......................... 67

### 3.2.3 The Scrum Framework - core practices ................................... 68

### 3.2.4 The relationship between Scrum practices and NPD characteristics ................................................................. 72

3. 3 Supplementary Scrum Practices ..................................................... 72

### 3.3.1 General Considerations ............................................................ 73

### 3.3.2 Sprint planning ..................................................................... 76

### 3.3.3 Sprint ..................................................................................... 78

### 3.3.4 Post-Sprint ............................................................................. 81

### 3.3.5 Release Management ............................................................. 82

3. 4 Summary ...................................................................................... 86
CHAPTER 7 - RESEARCH DETAILED DESIGN........................................................................ 183

7.1 INTRODUCTION ........................................................................................................... 183
7.2 MAJOR DESIGN INFLUENCES ..................................................................................... 184
  7.2.1 Fieldwork: Researcher involvement level .......................................................... 184
  7.2.2 Fieldwork: Data Access Management ............................................................... 186
  7.2.3 Design approach: Single v Multiple Case Study .............................................. 187
7.3 DETAILED DESIGN STAGE 1 - STUDY SETUP ......................................................... 187
  7.3.1 Management of Research Components .......................................................... 188
  7.3.2 Theory Development ......................................................................................... 191
7.4 DETAILED DESIGN STAGE 2 - DATA COLLECTION .................................................. 193
7.5 DETAILED DESIGN STAGE 3 - DATA ANALYSIS .................................................... 195
  7.5.1 Data Reduction .................................................................................................. 196
  7.5.2 Data Display ..................................................................................................... 197
  7.5.3 Conclusion and verification .............................................................................. 198
7.6 REFLECTIONS ON DESIGN APPROACH ................................................................. 199
7.7 CASE BACKGROUND................................................................................................ 200
  7.7.1 Background to the the I-E-S organization ....................................................... 201
  7.7.2 Background to the application of Scrum within I-E-S ...................................... 203
7.8 SUMMARY ................................................................................................................. 206

CHAPTER 8 - DISTRIBUTED SCRUM OF SCRUMS FINDINGS ........................................... 207

8.1 INTRODUCTION ........................................................................................................... 207
8.2 FIN-NN: DISTRIBUTED SCRUM OF SCRUMS CONTEXT ........................................ 208
  8.2.1 Background to the FIN-NN Project ................................................................. 209
  8.2.2 Product .............................................................................................................. 212
  8.2.3 Processes .......................................................................................................... 213
  8.2.4 Practices, Tools, Techniques ............................................................................. 214
  8.2.5 People ................................................................................................................ 216
8.3 FINDINGS ASSOCIATED WITH NEW PRODUCT DEVELOPMENT CHARACTERISTICS 217
  8.3.1 General considerations ..................................................................................... 218
  8.3.2 Pre-sprint .......................................................................................................... 220
  8.3.3 Sprint ................................................................................................................ 223
  8.3.4 Post-Sprint ........................................................................................................ 225
  8.3.5 Summary of NPD Findings .............................................................................. 226
8.4 FINDINGS ASSOCIATED WITH AGILE SOFTWARE DEVELOPMENT PRINCIPLES 229
  8.4.1 General Considerations .................................................................................... 230
  8.4.2 Pre-Sprint .......................................................................................................... 231
  8.4.3 Sprint ................................................................................................................ 233
  8.4.4 Post-Sprint ........................................................................................................ 235
  8.4.5 Summary of ASD Findings .............................................................................. 236
8.5 FINDINGS ASSOCIATED WITH GSD CHALLENGE AMELIATION GOALS ............. 239
  8.5.1 General considerations ..................................................................................... 240
  8.5.2 Pre-Sprint .......................................................................................................... 241
  8.5.3 Sprint ................................................................................................................ 243
  8.5.4 Post-Sprint ........................................................................................................ 247
  8.5.5 Summary of GSD Findings .............................................................................. 249
8.6 FINDINGS ASSOCIATED WITH LEAN SOFTWARE DEVELOPMENT VALUES .......... 252
  8.6.1 ‘V1’ Customer Value ........................................................................................ 253
  8.6.2 ‘V2’ Reduce Waste ........................................................................................... 256
  8.6.3 ‘V3’ Flow of Value ........................................................................................... 260
8.6.4 'V4' Person Focus ................................................................. 263
8.6.5 'V5' Continuous Improvement.............................................. 266
8.6.6 'V6' Product Excellence ........................................................ 268
8.6.7 'V7' Embrace Change ............................................................ 270
8.6.8 'V8' Business Environment Awareness .................................. 272
8.6.9 'V9' Data-driven Decisions .................................................... 273
8.6.10 'V10' Effective Process ......................................................... 276
8.6.11 'V11' Effective Collaboration ................................................. 280
8.6.12 'V12' Effective Use of Technology ......................................... 284
8.6.13 Summary of LSD Findings ..................................................... 286
8.7 Refined LdScrum Model (Scrum of Scrums) ................................. 290

CHAPTER 9 - TOTALLY INTEGRATED SCRUM FINDINGS ........................................ 294
9.1 Introduction ............................................................................. 294
9.2 Qual-NN Project: Totally Integrated Scrum Context ...................... 295
  9.2.1 Product .............................................................................. 297
  9.2.2 Processes .......................................................................... 297
  9.2.3 Practices, Tools, Techniques .................................................. 298
  9.2.4 People .............................................................................. 300
9.3 Findings Associated with New Product Development Characteristics .................................................. 301
  9.3.1 General considerations ........................................................ 301
  9.3.2 Pre-Sprint .......................................................................... 304
  9.3.3 Sprint ............................................................................... 307
  9.3.4 Post-Sprint ....................................................................... 310
  9.3.5 Summary of Npd Findings ..................................................... 311
9.4 Findings Associated with Agile Software Development Principles .................................................. 314
  9.4.1 General Considerations ........................................................ 314
  9.4.2 Pre-Sprint ........................................................................... 316
  9.4.3 Sprint ............................................................................... 318
  9.4.4 Post-Sprint ....................................................................... 320
  9.4.5 Summary of ASD Findings ..................................................... 322
9.5 Findings Associated with Gsd Challenge Alleviation Goals ............. 324
  9.5.1 General considerations ........................................................ 324
  9.5.2 Pre-Sprint ........................................................................... 326
  9.5.3 Sprint ............................................................................... 327
  9.5.4 Post-Sprint ....................................................................... 329
  9.5.5 Summary of GSD Findings ..................................................... 330
9.6 Findings Associated with Lean Software Development Values .......... 333
  9.6.1 'V1' Customer Value .......................................................... 334
  9.6.2 'V2' Reduce Waste .............................................................. 336
  9.6.3 'V3' Flow of Value .............................................................. 338
  9.6.4 'V4' Person Focus ............................................................... 340
  9.6.5 'V5' Continuous Improvement .............................................. 344
  9.6.6 'V6' Product Excellence ....................................................... 346
  9.6.7 'V7' Embrace Change .......................................................... 347
  9.6.8 'V8' Business Environment Awareness .................................. 348
  9.6.9 'V9' Data-driven Decisions .................................................... 350
  9.6.10 'V10' Effective Process ....................................................... 350
  9.6.11 'V11' Effective Collaboration ................................................. 354
  9.6.12 'V12' Effective Use of Technology ......................................... 359
  9.6.13 Summary of LSD Findings ..................................................... 361
List of Figures

Figure 1-1: Instances of Lean & Agile Thinking Philosophies .................................................. 2
Figure 1-2: Aggregate Nature of LT Instances: LM Contained by AM Whereas LSD Contains ASD ............................................. 9
Figure 1-3: Outline of Research Areas ....................................................................................... 11
Figure 1-4: Spectrum of Software Development Approaches (Adapted from Boehm, 2002) .................................................... 18
Figure 1-5: Application of Method Rationale Analysis Framework to LSD and Distributed Scrum ............................................. 31
Figure 1-6: High-Level Research Approach .............................................................................. 31
Figure 1-7: Work Breakdown Structure of Research Study ....................................................... 33
Figure 2-1: LSD Values WBS Segment ..................................................................................... 38
Figure 2-2: Influences of LSD ................................................................................................. 39
Figure 2-3: Evolution of Toyota Production System (TPS) (UL, 2005) ............................................. 39
Figure 2-4: The Toyota Production System House (Liker and Morgan, 2006) ............................................. 41
Figure 3-1: Scrum Practices WBS Segment .................................................................................. 62
Figure 3-2: Scrum Framework (Schwaber, 1995) ...................................................................... 64
Figure 3-3: Scrum Framework (Deemer et al., 2010) ................................................................. 69
Figure 4-1: Scrum in GSD WBS Segment ................................................................................... 89
Figure 5-1: Method Rationale Analysis WBS Segment .............................................................. 108
Figure 5-2: Framework for ISD Method Use (Fitzgerald et al., 2002) ........................................ 111
Figure 5-3: ISD Method Used and Developer Experience (Fitzgerald et al., 2002) ............................................. 113
Figure 5-4: Method Incorporating Process, Notation and Concepts (Adapted From Goldkuhl et al., 1998) ............................................. 114
Figure 5-5: Method Rationale (Ågerfalk and Wistrand, 2003) ............................................... 116
Figure 5-6: Levels of Method Abstraction (Ågerfalk & Fitzgerald, 2006) ...................................... 117
Figure 5-7: Method Rationale Analysis Framework (Ågerfalk, 2006) ...................................... 118
Figure 5-8: Potential LSD Value Base ..................................................................................... 122
Figure 5-9: ASD Goal Clusters (Ågerfalk, 2006) - Goal Codes Adapted to Reflect DScrum Goal Identifiers from Table 5-2 ............................................. 126
Figure 5-10: Goal Achievement Hierarchy for Internally Oriented DScrum Goals (Adapted From Ågerfalk, 2006) .......... 128
Figure 5-11: Values to Goal Mapping - LDSRUM Value Rationale - External Orientation .................. 131
Figure 5-12: Value to Goals Mapping - LDSRUM Value Rationale - Internal Orientation .................. 133
Figure 5-13: Pre-Sprint Method Rationale. ‘LDSRUM - Pre-Sprint’ (Fundamental View) ................. 140
Figure 5-14: Sprint Method Rationale. ‘LDSRUM - Sprint’ (Fundamental View) .................. 143
Figure 5-15: Post-Sprint Method Rationale. ‘LDSRUM - Post-Sprint’ (Fundamental View) .......... 145
Figure 6-1: Research Approach WBS Segment (Research Philosophy Work Package) ...................... 147
Figure 6-2: Summary of the Three Worlds (Mingers, 2001) ............................................................ 153
Figure 6-3: Framework for Organizational Multi-Disciplinary Research (Braa and Viden, 1999) ........ 165
Figure 6-4: Appropriate Methods for Different Approaches (Adapted from Braa and Viden, 1999) ........ 172
Figure 6-5: Iterative Research Approach (Adapted from Yin, 2009) ........................................ 175
Figure 7-1: Research Approach WBS Segment (Detailed Design & Case Context Work Packages) .......... 183
Figure 7-2: Data Analysis Process .......................................................................................... 195
Figure 7-3: Overall Iterative Process (Reflecting Hermeneutic Circle) ............................................ 198
Figure 7-4: Case Study Context (Adapted From Petersen and Wohlin (2009)) .......................... 202
Figure 8-1: Conduct Empirical Research WBS Segment .......................................................... 207
Figure 8-2: Fin-Link Team in Dedicated Scrum Meeting Room Using JIRA ................................. 212
Figure 9-1: Conduct Empirical Research WBS Segment ............................................................. 294
Figure 10-1: Report Empirical Research WBS Segment ............................................................... 367
Figure 10-2: Overall Pre-Sprint Case Goal Rationale ................................................................. 373
Figure 10-3: Comparison of Findings Counts in Sprint Planning (CS-4) Practice ....................... 373
Figure 10-4: Refined LDSRUM for Pre-Sprint Method Fragment ................................. 374
FIGURE 10-5: OVERALL SPRINT CASE GOAL RATIONALE .......................................................................................................................... 375
FIGURE 10-6: CONTRAST OF SCRUM - GENERAL ACTIVITIES (CS-7) .................................................................................................................. 375
FIGURE 10-7: REFINED LDSCRUM FOR SPRINT METHOD FRAGMENT ............................................................................................................ 376
FIGURE 10-8: POST SPRINT OVERALL CASE GOAL RATIONALE ..................................................................................................................... 377
FIGURE 10-9: COMPARISON OF FINDING COUNTS FOR POST-SPRINT FRAGMENT PRACTICES (CS10 & CS-11) .................. 377
FIGURE 10-10: REFINED LDSCRUM MODEL FOR POST-SPRINT METHOD FRAGMENT ...................................................................................... 378
FIGURE 10-11: BREAKOUT OF LSD VALUES FOR EACH PROJECT ................................................................................................................... 379
List of Tables

| Table 1-1: Characteristics of Lean Thinking and Agile Thinking in the Operations Sector (Naylor et al., 1999; Narasimhan et al., 2006; Naim and Gosling, 2011) | .............................................................. | 6 |
| Table 1-2: Similarities and Distinctions between LSD and ASD | .............................................................. | 8 |
| Table 2-1: Lean Principles from Various Sectors/Autors | .............................................................. | 53 |
| Table 2-2: Lean Terminology Applied to LSD | .............................................................. | 57 |
| Table 2-3: LSD Values and Associated Principles | .............................................................. | 60 |
| Table 3-1: New Product Development Principles That Inspired Scrum Founders (Takeuchi and Nonaka, 1986; Schwaber, 1995) | .............................................................. | 66 |
| Table 3-2: Agile Principles (Fowler, 2001) | .............................................................. | 68 |
| Table 3-3: Core Scrum Practices (Schwaber 1995; Highsmith 2002; Sutherland and Schwaber 2011) | .............................................................. | 71 |
| Table 3-4: Supplementary Scrum Practices - General Considerations (Sutherland, 2006; Deemer et al., 2010; Cohn, 2010) | .............................................................. | 75 |
| Table 3-5: Supplementary Scrum Practices - Initial Planning and Sprint Planning (Sutherland, 2008; Deemer et al., 2010; Cohn, 2010) | .............................................................. | 77 |
| Table 3-6: Supplementary Scrum Practices - Sprint (Cohen et al., 2004; Sutherland, 2008; Cohn, 2010) | .............................................................. | 79 |
| Table 3-7: Supplementary Scrum Practices - Technical Practices (Cohen et al., 2004; Cohn, 2010) | .............................................................. | 80 |
| Table 3-8: Supplementary Scrum Practices - Post-Sprint (Cohen et al., 2004; Sutherland, 2006; Deemer et al. 2010) | .............................................................. | 82 |
| Table 3-9: Supplementary Scrum Practices - Release (Cohen et al., 2004; Deemer et al., 2010; Gannon, 2011) | .............................................................. | 83 |
| Table 3-10: Summary of Scrum Practices Related to NPD and ASD Principles | .............................................................. | 85 |
| Table 3-11: Synthesis of Guiding Principles from Supplementary Practices to Core Set of Practices | .............................................................. | 86 |
| Table 4-1: GSD Challenges and Benefits (Ágerfalk et al., 2005) | .............................................................. | 91 |
| Table 4-2: Goals Derived from GSD Challenges (Ágerfalk et al., 2005) | .............................................................. | 92 |
| Table 4-3: General Considerations - GSD Scrum Practices (Hossain et al., 2011; Cohn, 2010; Dodda & Ansari, 2010) | .............................................................. | 95 |
| Table 4-4: Initial Planning - GSD Scrum Practices (Hossain et al., 2011; Cohn, 2010; Dodda & Ansari, 2010) | .............................................................. | 96 |
| Table 4-5: Pre-Sprint GSD Scrum Practices (Hossain et al., 2011; Cohn, 2010; Dodda & Ansari, 2010) | .............................................................. | 96 |
| Table 4-6: Sprint GSD Practices (Holmstrom, 2006; Cristal et al., 2008; Hossain et al., 2011; Possiara et al., 2009; Cohn, 2010; Dodda & Ansari, 2010; Woodward, 2010) | .............................................................. | 99 |
| Table 4-7: Post-Sprint GSD Practices (Hossain et al., 2011; Cohn 2010; Dodda & Ansari 2010) | .............................................................. | 100 |
| Table 4-8: GSD Scrum Team Configuration Practices (Sutherland et al., 2007) | .............................................................. | 103 |
| Table 4-9: Summary of Scrum Practices Related to GSD Challenges | .............................................................. | 105 |
| Table 4-10: Synthesis of Goals That Overcome GSD Challenges from Supplementary Scrum GSD Practices to Core Set of Scrum Practices | .............................................................. | 106 |
| Table 5-1: Dimensions of Method Fragments Used in Situational Method Assembly (Brinkkemper et al., 1998) | .............................................................. | 114 |
| Table 5-2: DS/Scrum - Goal to Principle Mapping | .............................................................. | 125 |
| Table 6-1: Summary of 'soft' v 'hard' research dichotomies (Adapted from Fitzgerald and Howcroft, 1998) | .............................................................. | 171 |
| Table 6-2: Description of overall interpretive research - adapted from (Klein and Myers, 1999) | .............................................................. | 181 |
| Table 7-1: Summary of solutions to fieldwork issues reported in Walsham (2006) | .............................................................. | 186 |
| Table 7-2: Requirements, propositions and hypotheses of this case study | .............................................................. | 189 |
| Table 7-3: Formal Scrum Adoption Program in I-E-S | .............................................................. | 206 |
| Table 8-1: Configuration of I-E-S R&D Financials Project Scrum Development Team | .............................................................. | 211 |
| Table 8-2: Candidate NPD goals pursued by the ‘General Considerations’ fragment (from table 3-11) | .............................................................. | 218 |
| Table 8-3: Candidate NPD goals pursued by practices in ‘Pre-Sprint’ method fragment (from table 3-11) | .............................................................. | 220 |
| Table 8-4: Candidate NPD goals pursued by practices in ‘Sprint’ fragment (adapted from table 3-11) | .............................................................. | 223 |
| Table 8-5: Candidate NPD goals pursued by practices in ‘Post Sprint’ method fragment (from table 3-11) | .............................................................. | 225 |
| Table 8-6: Overview of field results related to the presence of NPD goals | .............................................................. | 228 |
Table 8-7: Candidate ASD goals pursued by practices in 'General Considerations' fragment (adapted from Table 3-11) ................................................................. 230
Table 8-8: Candidate ASD goals pursued by practices in 'Pre-Sprint' method fragment (from Table 3-11) .................................................... 231
Table 8-9: Candidate ASD goals pursued by practices in 'Sprint' method fragment (from Table 3-11) .................................................. 233
Table 8-10: Candidate ASD goals pursued by practices in 'Post-Sprint' method fragment (adapted from Table 3-11) .................................................................................................................. 235
Table 8-11: Overview of field results related to the presence of ASD goals ................................................................. 238
Table 8-12: Candidate GSD Challenge Alleviation goals pursued by practices in 'General Considerations' method fragment (adapted from Table 4-10) .................................................................................. 240
Table 8-13: Candidate GSD Challenge Alleviation goals pursued by practices in 'Pre-Sprint' method fragment (adapted from Table 4 10) .................................................................................................................. 241
Table 8-14: Candidate GSD Challenge Alleviation goals pursued by practices in 'Sprint' method fragment (adapted from Table 4 10) .................................................................................................................. 243
Table 8-15: Candidate GSD Challenge Alleviation goals pursued by practices in 'Post-Sprint' method fragment (adapted from Table 4 10) .................................................................................................................. 247
Table 8-16: Simplistic view showing field results for presence of GSD Challenge Alleviation goals .................................................. 251
Table 8-17: Summary of candidate LSDScrum value rationale .............................................................................................................................. 253
Table 8-18: Candidate Value Rationale for V1 - 'Customer Value' .......................................................................................................................... 253
Table 8-19: Candidate Value Rationale for V2 - 'Reduce Waste' ......................................................................................................................... 256
Table 8-20: Candidate Value Rationale for V3 - 'Flow of Value' ......................................................................................................................... 260
Table 8-21: Candidate Value Rationale for V4 - 'Person Focus' ......................................................................................................................... 263
Table 8-22: Candidate Value Rationale for V5 - 'Continuous Improvement' ........................................................................................................ 266
Table 8-23: Candidate Value Rationale for V6 - 'Product Excellence' .................................................................................................................. 268
Table 8-24: Candidate Value Rationale for V7 - 'Embrace Change' .................................................................................................................... 270
Table 8-25: Candidate Value Rationale for V8 - 'Business Environment Awareness' .......................................................................................... 272
Table 8-26: Candidate Value Rationale for V9 - 'Data-driven Decisions' ........................................................................................................ 273
Table 8-27: Candidate Value Rationale for V10 - 'Effective Process' ............................................................................................................ 276
Table 8-28: Candidate Value Rationale for V11 - 'Effective Collaboration' ....................................................................................................... 280
Table 8-29: Candidate Value Rationale for V12 - 'Effective Use of Technology' .............................................................................................. 284
Table 8-30: Overview of all LSD findings ...................................................................................................................................................... 289
Table 8-31: Refined Goal Rationale (Scrum of Scrum findings) ..................................................................................................................... 292
Table 8-32: Refined LSD Value Rationale (Scrum of Scrum findings) ........................................................................................................ 293
Table 9-1: Configuration of I-E-S R&D QUAL Scrum development team ...................................................................................................... 300
Table 9-2: NPD goals pursued by practices in 'General Considerations' fragment (adapted from Table 3-11) 301
Table 9-3: NPD goals pursued by practices in 'Pre-Sprint' method fragment (adapted from Table 3-11) .................. 304
Table 9-4: NPD goals pursued by practices in 'Sprint' method fragment (adapted from Table 3-11) ................................. 307
Table 9-5: NPD goals pursued by practices in 'Post-Sprint' method fragment (adapted from Table 3-11) ................................. 310
Table 9-6: Overview of field results related to presence of NPD goals ......................................................................................................... 313
Table 9-7: ASD goals pursued by practices in 'General Considerations' fragment (adapted from Table 3-11) 314
Table 9-8: Candidate ASD goals pursued by practices in 'Pre-Sprint' method fragment (from Table 3-11) 316
Table 9-9: Candidate ASD goals pursued by practices in 'Sprint' method fragment (from Table 3-11) ................................. 318
Table 9-10: Candidate ASD goals pursued by practices in 'Post-Sprint' method fragment (adapted from Table 3-11) .................................................................................. 320
Table 9-11: Overview of field results related to presence of ASD goals ......................................................................................................... 323
Table 9-12: Candidate GSD Challenge Alleviation goals pursued by practices in 'General Considerations' method fragment (adapted from Table 4-10) .................................................................................. 324
Table 9-13: Candidate GSD Challenge Alleviation goals pursued by practices in 'Pre-Sprint' method fragment (adapted from Table 4 10) ................................................................................................. 326
Table 9-14: Candidate GSD Challenge Alleviation goals pursued by practices in 'Sprint' method fragment (adapted from Table 4 10) ................................................................................................. 327
Glossary of Acronyms

AM  Agile Manufacturing
APM  Agile Project Management
ASD  Agile Software Development
AT   Agile Thinking
GSD  Global Software Development
LD   Lean Development
LM   Lean Manufacturing
LPD  Lean Product Development
LSD  Lean Software Development
LT   Lean Thinking:
PM   Project Management
SCM  Supply Chain Management
TPPD Toyota Product Development System
TPS  Toyota Production System

Glossary of LDScrum Terms

LDScrum  Lean Distributed Scrum: Proposed framework linking LSD values that underpin goals pursued through the application of Scrum practices.

DScrum  Distributed Scrum: Subset of LDScrum. This portion of the framework links the identified DScrum goals with core Scrum practices.

Method Fragment  All or part of an ISD method. In the context of the LDScrum model, three method fragments of the Scrum method are presented. The three fragments are pre-sprint; sprint and post-sprint and each fragment is described as containing a number of core Scrum practices.

Practice Core Scrum Practice  The Scrum method is presented as consisting of twelve core practices. The LDScrum model presents 9 of these practices in three method fragments.

Practice Lean Principle  Lean principles are identified from literature reviews of the application of lean concepts to operations management, product development, project management and software development. These principles are used to induce a set of LSD values that form the third pillar of the LDScrum model.
ASD Principle

ASD principles are twelve agile software development principles proposed by the signatories of the Agile Manifesto. These guidelines are interpreted as goals to be pursued by agile methods and as such form part of the DScrum goal-set that constitutes the second pillar of the LDScrum model.

Goal

DScrum Goal

Twenty-one DScrum goals form the 2nd pillar of the LDScrum model. This goal-set is made up of NPD, ASD and GSD goals. It acts as the link between core Scrum practices and LSD values and as such enables a view of the relationship between LSD and one particular instance of ASD (Scrum in GSD).

Value

LSD Value

A general belief or understanding about lean software development that influences developers on the development goals they pursue. A set of LSD values are induced from lean principles and this set forms the third pillar of the LDScrum model.

Value Base

LSD Value base

An actor (or developer) leverages a value base in order to determine which goals are worth pursuing.

Value Rationale

LSD Value rationale

Goals are anchored by one or more values. One value may anchor multiple goals.

The LDScrum value rationale presents each LSD value and the goals that it is deemed to underpin. The candidate LDScrum model shows the associations gleaned from analysis of the value and goal descriptions. The refined LDScrum model shows the rationale from findings of why certain goals appeared to be pursued. LDScrum value rationale links LSD to ASD.

Value Hierarchy

Values may be anchored by higher-order values. It is also possible that values may contradict other values within an actor’s value base.

Goal Rationale

LSD Goal rationale

Method fragments exist to achieve one or more goals. A goal may be achieved by multiple method fragments.

The DScrum goal rationale represents the association between core Scrum practices and DScrum goals. The candidate model is built from the associations of supplementary Scrum practices to the 21 NPD, ASD and GSD goals. Ultimately, this enables a linkage between practices and LSD values. These rationales are presented separately for each method fragment.

Goal Hierarchy

Goals may be pursued to achieve higher-order goals. It is also possible that pursuit of a particular goal may contradict the achievement of another goal.
Chapter 1- Introduction

1.1 Motivations for study

Information Systems Development (ISD) may be conducted using a variety of work approaches ranging from chaotic ‘hacking’ through to prescriptive predictable contract-driven methods (Fitzgerald et al., 2002; Boehm, 2002; Avison and Fitzgerald, 2003). Agile software development (ASD) has emerged as a practice-led paradigm\(^1\) which offers great promise in improving software productivity. However some confusion exists as to its relationship with Lean Software Development (LSD). Some treat LSD as more or less synonymous with ASD whereas others view LSD as a different paradigm. The definition and positioning of LSD relative to ASD is important as it gets to the heart of software development as craft versus science debate. The purpose of this research study is to present a deeper understanding of the relationship between these two paradigms.

The view taken in this thesis is that LSD is a broad management philosophy and its relationship to ASD is examined through an association of LSD values to a specific application of one ASD method. Sub-sections 1.1.1 to 1.1.6 use views from literature to describe the motivations for this study. Commentary on the application of lean and agile concepts in different domains illustrates the confusion associated with lean and agile terminology and explains the overlapping nature of LSD and ASD. Identified gaps in ASD research are noted as drivers for the selection of the ASD instance used to focus this broad research objective.

1.1.1 Philosophical foundations of LSD and ASD

A prerequisite to an effective examination of LSD is the formation of a clear understanding of the philosophy that underpins this software development paradigm:

\(^1\) The term “paradigm” here is intended to refer to a model or set of ideas that forms the basis for an ISD work approach.
‘lean thinking’. A definition of the relationship between lean thinking (LT) and agile thinking (AT) will help to form a basis for exploration of the relationship between the application of these two philosophies in the field of software development.

Both LT and AT are considered to be philosophies as they present the guiding values or beliefs that inform different implementations of work activities. Figure 1-1 presents some paradigms that have been influenced by LT and AT. It should be noted at this point that this diagram makes the assumption that LT is applied in the LSD paradigm and that AT is applied in the ASD paradigm. This assumption may be misleading and consequently serve to add to the confusion around these two paradigms. The following paragraphs review the higher-order philosophies of LT and AT and then drill into their related paradigms in order to expose various inconsistencies that hinder clarity on this subject.

![Figure 1-1: Instances of Lean & Agile thinking philosophies](image)

Confusion exists among both practitioners and academics as to the meaning of LT as a result of the evolution of this philosophy from its origins in the automotive manufacturing sector through its dissemination in different environments. For example, initial focus on shop floor practices led to a misunderstanding that value creation was
merely an outcome of cost reduction resulting from waste removal. Lean is characterised as “doing more with less” and is often associated with the manufacturing concept of ‘zero inventory’ in order to perform just-in-time (JIT) flow of work (Towill and Christopher, 2002). In the 1990’s, such views were expanded to embrace quality and other considerations evoked by Womack and Jones (1996) in their proposal of certain LT principles. The principle of ‘value’ sharpened the definition of value creation to be the requirement to create value as it is perceived by the customer. Another principle, ‘value stream’ emphasized the concept of supply-chain management and the extension of manufacturing concerns beyond the walls of the production plant. Major gaps that have been highlighted throughout the evolution of LT include an over-emphasis on the automotive sector, sub-optimization of the entire system through over-emphasis on the shop floor, lack of appreciation for human factors, and a simplistic approach to variability management. The evolution of LT could be described as a movement along a continuum from tactical lean initiatives on the shop floor to a strategic contingency based customer-orientation (Hines et al., 2004). As a result, the application of LT exists at two levels:

- an operational level (most evident in the shop floor tools and techniques that have been employed with much success at Toyota (Ohno, 1988)).
- A strategic level that embraces a major focus on the creation of value and thorough understanding of customer-defined value.

In non-repetitive high-volume manufacturing environments, the application of LT has encountered challenges in addressing issues such as volatility and variability, resulting in proposals that agile or hybrid approaches may be more appropriate to these situations (Hines et al., 2004).

In order to address the fact that no universal definition existed for an ASD method due to a lack of clear grounding with AT, Conboy and Fitzgerald (2004) proposed a definition for AT that could be used for this purpose. The construction of this definition from the literature incorporated the philosophies of flexibility and LT, thus proposing that LT is a
component of AT. The application of AT enables an organization to embrace change and either create or effectively respond to change both internally and externally.

### 1.1.2 Application of LT and AT to manufacturing: LM and AM

The muddled nature of the relationship between LT and AT is further seen by considering instances of application of these philosophies. A review of studies into the relationship between ‘Lean manufacturing’ (LM) and ‘Agile manufacturing’ (AM) reflected the aforementioned operational to strategic continuum (Narasimhan et al., 2006). It proposed that these are two distinct overlapping paradigms where a temporal relationship exists that may reflect upon the capability of an organization. An organization may realize performance improvements using LM due to reduced waste. However in order to increase its capacity to respond to changing market conditions, it needs to ‘evolve’ to a higher capability reflected in the AM paradigm.

Furthermore, the ‘murky’ nature of reporting of sets of practices and performance outcomes associated with these two paradigms is noted as another source of confusion. They may share a common subset of practices. Notable LM practices include the creation of JIT flow, development of human resources, effective management of equipment and quality control. AM utilizes advanced manufacturing technologies. It also emphasizes employee skills development, supplier alliances and customer relationships. As a result, they are explored via a deeper reflection on concepts of ‘leaness’ and ‘agility’ (The word ‘leaness’ used in the review is derived from Womack and Jones (2003) and as such is considered to be derived from the term ‘lean thinking’ used above). Leaness is concerned with performance improvements in cost efficiency, reliability, conformance to quality standards and delivery speed. Agility promotes performance improvements through increased responsiveness. A consensus view among authors has emerged that operational leaness is an antecedent of strategic agility. Thus production techniques may be viewed as a chronology of craft/mass/lean and finally AM (Narasimhan et al., 2006).
1.1.3 Application of LT and AT to Supply Chain Management (using LM and AM)

The overlapping nature of LM and AM noted above is also seen in a second instance of the application of LT and AT: operations of supply chain management (SCM) (Naylor et al., 1999). A hybrid approach to SCM is proposed that employs LM to upstream activities from a ‘decoupling point’ (where customer demand influenced work) and AM for subsequent downstream activities. This ‘leagile’ approach effectively combines the reliance of LM on levelling demand (to enable flow) with the responsive nature of AM (to meet changing market needs) (Narasimhan et al., 2006).

The hybrid approach that involves the combination of both LT and AT within SCM can result in the application of these approaches at different times in the same place or at the same time in different places. This is necessary in order to accommodate different situations where cost may be top priority in which case a lean approach is most effective as against instances where product availability or service level is the key to winning the market in which case the most desirable approach is agile (Towill and Christopher, 2002). LT and AT may be evident in overlapping manufacturing paradigms (LM and AM) or be considered in the application of different approaches to SCM. In order to distinguish these two philosophies, table 1-1 synthesizes views on the characteristics of the application of LT and AT to the operations sector (manufacturing plants and SCM).
<table>
<thead>
<tr>
<th>Dimension</th>
<th>Lean Thinking</th>
<th>Agile Thinking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of market knowledge</td>
<td>Essential</td>
<td>Essential</td>
</tr>
<tr>
<td>Value stream</td>
<td>Essential</td>
<td>Essential</td>
</tr>
<tr>
<td>Reduction of lead time</td>
<td>Essential</td>
<td>Essential</td>
</tr>
<tr>
<td>Waste elimination</td>
<td>Essential</td>
<td>Desirable</td>
</tr>
<tr>
<td>Rapid reconfiguration</td>
<td>Desirable</td>
<td>Essential</td>
</tr>
<tr>
<td>Smooth demand/level schedules</td>
<td>Essential</td>
<td>Arbitrary</td>
</tr>
<tr>
<td>Ability to deal with variable production requirements</td>
<td>Arbitrary - this is addressed by levelling the schedule using market knowledge and forward planning</td>
<td>Essential</td>
</tr>
<tr>
<td>Key performance indicators</td>
<td>Lead time, Quality and Cost</td>
<td>Lead time, Quality and level of service</td>
</tr>
<tr>
<td>Planning Horizon</td>
<td>Initially was tactical but has evolved to embrace strategic concerns</td>
<td>Strategic - emphasising customer linkages and market needs</td>
</tr>
<tr>
<td>Capability growth of manufacturer organization</td>
<td>Increased capability from mass production only</td>
<td>Increased capability from mass and lean.</td>
</tr>
</tbody>
</table>

Table 1-1: Characteristics of Lean Thinking and Agile Thinking in the operations sector (Naylor et al., 1999; Narasimhan et al., 2006; Naim and Gosling, 2011)

### 1.1.4 Application of LT and AT to Software development (LSD & ASD)

Having reviewed LT and AT, LM and AM and the application of these paradigms in supply chain management, further ambiguity is evident in LSD and ASD. Fitzgerald et al. (2002) define an information systems development (ISD) method as ‘a coherent and systematic approach’ that guides developers on the steps to be performed to construct a system. Guidelines may include the rationale for the application of particular work activities based upon a particular philosophy of systems development. The notion of ‘systems development philosophy’ refers to the main concerns being addressed by the development organization e.g. maintainability, security, system reliability, end-user involvement in system construction, speed of development etc. It seems reasonable to assume that ASD methods would pursue a philosophy of agility i.e. the characteristics of AT would be evident in ASD methods. Many such methods emerged to counteract the perceived weaknesses of approaches that were based on the traditional systems development lifecycle or ‘Waterfall’ model. These ‘lightweight’ methods sought to avoid
wasteful activities in software development. Research is ongoing to determine the effectiveness of these methods (Abrahamsson et al., 2009).

One particular ‘lightweight’ method was named ‘Lean development’ (LD) (Cohen et al., 2004; Charette, 2003; Highsmith, 2002). This is considered a top-down strategic approach to software development in contrast to other agile methods (Highsmith, 2002). The application of LT to software development approaches was also reported under the labels: ‘lean hardware/software development’ (Hou, 1995) and ‘lean software development’ (Middleton, 1995; Middleton, 2001a; Middleton et al., 2005; Middleton and Joyce, 2012; Raman, 1998; Poppendieck and Poppendieck, 2003). This thesis uses the term LSD as an umbrella term to refer to software development methods that claim to apply LT.

Confusion exists about whether LSD is indeed an agile method or a different form of development that has a close relationship to ASD. In their roadmap for research into ASD, Dingsoyr et al. (2008) indicate that very little research has been conducted on LSD. Various reviews of agile methods have included Charette’s LD method whereas others have omitted it. Indeed, Charette’s LD method advocates a strategic perspective and promotes ‘change tolerance’ which appears to be strongly aligned with AT and yet has become synonymous with LT (or at least ‘lean’, possibly due to its name). This adds to the muddled nature of the relationship between LSD and ASD. Furthermore, various agile methods recommend practices and techniques that support the achievement of lean principles and goals (Hibbs et al., 2009). The view is taken in this study that LSD is a broader paradigm that considers software development from an overall business perspective. It denotes software development approaches that adhere to key lean principles. Fowler (2008) maintains that development teams should not consider whether to perform LSD or ASD. Instead they should aim to meet the principles of LT in the application of an agile method - “developers should not do agile or lean - they should do agile and lean” (Fowler, 2008, para. 9). Table 1-2 contrasts LSD with ASD along a set of
dimensions in order to emphasize the similarities and distinctions between these two approaches.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>LSD</th>
<th>ASD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary audience</td>
<td>Management</td>
<td>Developers</td>
</tr>
<tr>
<td>Leadership style</td>
<td>Team Management (Blake &amp; Mouton, 1964)</td>
<td>Team Management</td>
</tr>
<tr>
<td>Planning horizon</td>
<td>Strategic</td>
<td>Tactical</td>
</tr>
<tr>
<td>Communication</td>
<td>Throughout enterprise (value stream)</td>
<td>At local project level</td>
</tr>
<tr>
<td>Process structure</td>
<td>Iterative production of development</td>
<td>Iterative production of development</td>
</tr>
<tr>
<td></td>
<td>increments - iterations may be flow-based (Kanban) or time-boxes</td>
<td>increments - iterations are time-boxed.</td>
</tr>
<tr>
<td>Design philosophy</td>
<td>Consider long-term view and promote</td>
<td>Keep design simple and only build for</td>
</tr>
<tr>
<td></td>
<td>change-tolerant designs</td>
<td>current need</td>
</tr>
<tr>
<td>Value</td>
<td>Enterprise performance in meeting</td>
<td>Project performance in meeting</td>
</tr>
<tr>
<td></td>
<td>customer’s needs</td>
<td>customer’s needs</td>
</tr>
<tr>
<td>Process improvement</td>
<td>Rigorous PDCA cycle (Shalloway et al., 2010)</td>
<td>Informal inspect &amp; adapt approach</td>
</tr>
</tbody>
</table>

Table 1-2: Similarities and distinctions between LSD and ASD

1.1.5 Inconsistencies between LT and AT application

In reviewing two potentially related concepts, A and B, there are four relationships to be considered: A and B are unrelated, A and B are separate but partially related, A is a subset of B or B is a subset of A. In reviewing LT and AT above, it is noted that these are two distinct but related concepts. Evidence of this relationship is highlighted by many similarities in instances of their application in the contexts of manufacturing and software development. However, the inconsistency of some of the differences in these applications highlights reasons for confusion. LM is considered to exist (at least partially) at the operational level whereas AM has a strategic focus in its consideration of changing market needs. However this tactical/strategic focus is reversed when the software development context is explored: LSD emphasizes strategic thinking whereas ASD focuses on tactical considerations. Another remarkable inconsistency is that the aggregate nature of instances of LT and AT differs depending upon their application context. LM is reported to be an antecedent of AM and as such could be perceived as a subset of AM in the same way that mass manufacturing may be considered a subset of LM. On the other
hand, ASD is considered to be a tactical subset of LSD as the latter applies the broader management philosophy espoused by LT. Figure 1-2 illustrates this ‘flip-flop’ anomaly that adds to the confusion around the relationship between these areas.

![Diagram](image)

**Figure 1-2: Aggregate nature of LT instances: LM contained by AM whereas LSD contains ASD**

In summary, LT is reported as addressing both operational and strategic concerns, having evolved from initial operational waste-reduction initiatives to a more strategic outlook. AT is a strategic philosophy that embraces changes and incorporates the operational concerns of LT (so long as it does not impact on responsiveness to change). LM is an application of LT that is considered a tactical antecedent of the more strategic AM (an application of AT), although they may both be used to complement each other in supply chain management. Another application of LT, LSD is the opposite - it is considered a strategic approach that may use the more tactical ASD as part of its implementation. Finally, LD was considered lean due to its name, but yet was sometimes listed as an agile method and appeared to follow the more strategic considerations of AT. As stated at the outset of this chapter, the primary goal of this research study is to present a deeper understanding on the relationship between LSD and ASD.
1.1.6 **ASD method application used to explore LSD relationship**

Fitzgerald et al. (2002) state that a key characteristic of an ISD method is that it calls out for the presentation of a set of guidelines specifying the ‘what, why and how’ of the steps performed to develop a system. LSD does not specify these details and as such cannot be considered an ISD method. As stated above, the view is taken that LSD is an ISD paradigm reflecting a management philosophy rather than an ISD method and this is explored through a genealogical analysis of the origins of LSD. Various sources are examined in order to identify principles associated with the application of LT in different domains. A synthesis of these principles facilitates the derivation of a candidate set of lean values that characterise LSD. Although immediately valuable to practitioners seeking to apply lean values in agile projects, the primary reason for proposing this value-set is to assess elements of LT in the principles and practices of various agile methods so as to present a deep understanding of the relationship between LSD and ASD and facilitate optimal applicability of LT using agile methods.

In order to narrow the focus of such a broad objective, the research area is confined to specific aspects of the overall theme (figure 1-3):

- **GLOBAL SOFTWARE DEVELOPMENT (GSD).** Research is restricted to the management of teams whose members are based in different global locations.
- **PROJECT MANAGEMENT and SCRUM.** The application of ASD can cover many subjects such as design, quality management, requirements engineering, etc. This study concentrates on the project management of work carried out using ASD methods. In order to further narrow the focus of this research, it reports upon work carried out using one particular agile approach which is especially suited to the project management of software development situations: SCRUM.
LEAND SOFTWARE DEVELOPMENT. The research explores the application of SCRUM in a globally distributed context to determine the relationship of such a process with the principles of LT.

This restriction reduces the research goal to the presentation of a deeper understanding about the relationship between LSD and one particular ASD method when performed within GSD. There are two reasons for the selection of GSD as the vehicle to constrain the area of investigation:

- Relevance to industry
- Relevance to ASD research

In recent years, software development has been increasingly performed by distributed teams (O’Conchuir et al., 2009). Research has identified the potential for ‘waste’ to be
introduced into such development situations due to issues caused by temporal, geographical and cultural distance. By the same token, studies have proposed that ‘value’ is added to development efforts due to benefits realized from distributed team configurations (Ågerfalk et al., 2005; O’Conchuir et al., 2009).

Studies have identified a need for rigorous research to be applied to many aspects of ASD (Abrahamsson et al., 2009). As existing distributed teams begin to adopt agile methods, research into the application of agile in globally distributed environments has the potential to be useful to the developer community. Ågerfalk et al. (2009) conducted a Delphi study to identify future research topics needed in the areas of flexible and distributed information systems development. The application of ASD within globally distributed teams was identified as a major topic requiring further research:

“\textit{Distributed development recognizes that, more and more, ISD takes place in globally distributed settings.....empirical studies are sought that assess existing agile methodologies with respect to their impact on agility of distributed IS development teams.}”

(Ågerfalk et al., 2009, p. 323)

In summary, this study forms part of the work required to address a research gap in an area that is relevant to industry needs. It investigates the influence of LSD values on the application of an ASD method (Scrum) within globally distributed teams in order to present rigorous research on a specific application of agile development. A sound theoretical foundation is recommended for the effective execution of a research initiative (Levy and Ellis, 2006). To this end, the next section presents an overview of previous research deemed to be relevant to the areas identified above: GSD, ASD, Agile project management and LSD. Furthermore, literature produced in the areas of LSD, Scrum and the application of Scrum in GSD are addressed in detail in chapters 2, 3 and 4 respectively. This work is used to develop the conceptual framework used to plan, execute and report on the empirical research.
1.2 Overview of areas under research

This section introduces the main ISD aspects examined in the literature analysis and empirical research. The first and last sub-sections are necessarily brief and are merely introductions to the subject.

The first sub-section in this overview (1.2.1) presents a high-level introduction to LSD. This section contrasts LSD with ASD by describing how it has been addressed by various ASD-related publications. It then presents a high-level overview of the origins of LSD and describes some observations on the application of LT concepts to software development (LSD). The literature on LSD is reviewed in detail in chapter 2 and this work is used to develop a set of LSD values that form a core pillar of the conceptual framework used in the research.

The second research area presented is ASD. Section 1.2.2 positions ASD in the spectrum of ISD approaches and presents the values and characteristics of this form of software development. Section 1.2.3 narrows the ASD focus further by presenting literature on a particular aspect of ASD: project management. It concludes with a broad description of Scrum as this is the ASD project management method selected to filter the research further. The literature used to relate Scrum to this research is described in detail in chapter 3.

Finally, sub-section 1.2.4 introduces the concept of GSD. The high level nature of this introduction reflects the position that the overall study is not a broad examination of the application of GSD. This ISD approach is used as a lens to scope the main subject under investigation: the relationship between LSD and ASD. Literature on the specific details required to relate GSD application to ASD will be presented in chapter 4.
1.2.1 Lean Software Development

The term ‘lean software development’ (LSD) is not exclusively concerned with efficient code or ‘lean software’ (Wirth, 1995). Instead, it refers to the approach taken to the creation of software. Different opinions on whether LSD is an agile method are evident by the inclusion of Charette’s LD method in lists of agile methods in practice (Jalali and Wohlin, 2011; Dyba and Dingsoyr, 2008; Cohen et al., 2004), while other notable reviews of agile software methods omit it from their analysis (Abrahamsson et al., 2003; Lindvall et al., 2002). Although Cohen et al. (2004), included Charette’s LD in their review of various agile methods, they did not examine it as rigorously as other methods due to their interpretation of it as being more of a management philosophy than a software development process. Charette (2003) claimed that neither project-based methods nor software process improvement models have introduced the stability necessary to instil confidence in management as to the reliability of software engineering activities. He proposed that organizations needed to adopt a more holistic approach to software development. LD recommends various management strategies that enable the alignment of software development with an organization’s strategy.

A key figure in the creation and evolution of Scrum, (Sutherland, 2008) reflected upon this confusion in his musings on the relationship between Scrum and LSD. He contended that Scrum and LSD are two separate systems that emerged to address the issues of complex adaptive systems. He suggested that LSD provides a useful background to explain the advantages of Scrum implementations to business leaders and also to educate developers on the errors they may be making in their Scrum implementations. Other researchers have proposed that most agile methods are considered to be tactical whereas LSD is thought to have a strategic focus (Hibbs et al., 2009; Coplien and Bjornvig, 2010). These views would be consistent with the recommendations of Fowler (2008) that one should consider the application of “lean and agile” (cf. section 1.1.4).

LSD has emerged from concepts introduced over the past two centuries in business management, especially in the manufacturing sector. The ‘Toyota Production System’
(TPS) resulted from the development and application of lean practices at Toyota Motor Corporation since the mid 1940s. This system is cited by many as a key reference for parties interested in this subject. The core of this system is the pursuit of waste elimination and total respect for people (workers, customers, suppliers etc.) (Ohno, 1988).

The development of the TPS resulted from a series of iterative improvements and innovations to the organization, enabled by a culture of continual learning. The constant fluidity of the system due to the evolution of practices driven from worker feedback meant that the entire TPS was not published until twenty years after the program had begun. This has some resonance with the software development sector where practitioners have applied ‘situational methods’ or methods-in-action in order to adapt to the needs of their environment (Fitzgerald et al., 2002). Indeed, the publication of the Agile manifesto in Snowbird Utah, in 2001 was a reflection on ‘lightweight’ methods that had been applied for many years (Fowler and Highsmith, 2001).

Innovations practised at Toyota were widely disseminated in work produced by the MIT International Motor Vehicle Program (IMVP) - a research program studying the role of automobiles in the future (Holweg, 2007). This work was reported in Womack et al.,(1990). A promotional world-wide tour by the authors to launch the book revealed a lot of interest in their findings. However, it also highlighted that many practitioners were able to understand individual lean techniques but found it very challenging to adopt a holistic approach and understand how to ‘become lean’. As a result, Womack and Jones (2003) distilled many of the experiences and techniques that they encountered in their research and coined the phrase ‘Lean Thinking’ which proposed five principles of lean: Value, Value Stream, Continuous Flow, Pull and Perfection.

LT has diverged from being a manufacturing concept that introduces innovations into repetitive manufacturing processes. Liker and Morgan (2006) noted that although lean was being adapted in manufacturing plants world-wide, the company that has served as the core model for lean programmes, Toyota, had itself developed and used a different set
of lean principles to drive its product development organization. The Toyota Production Systems (TPS) is hailed as the foundation for LT, but the Toyota Product Development System has been used to evolve its creation of new products. The requirement to create new high-quality unique solutions is familiar to software developers. The Toyota Product-Process Systems (TPPD) consists of a set of 13 principles incorporating the areas of process, people and tools. Reinertsen and Shaeffer (2005) proposed that lean principles had an important place in R&D, but that it would be a mistake to apply lean principles to R&D in the same way that they are applied to manufacturing.

The principles of LT have been applied to different domains. In the software development industry, different approaches have been taken to the incorporation of lean principles (Ladas, 2009). Reinertsen (2009) warned against putting ‘faith’ before ‘science’ when he recommended that software developers should not rely on blind adherence to the application of lean principles to their craft. He maintained that the abstraction of many of the TPS lean principles reveals their basis in management science. He advised that congnisance of this field should be taken in the selection and application of those same principles and practices to software development. Middleton (2001b) used the work of Womack and Jones (2003) to comment on the application of LT to the software development process. One notable outcome of his research is the influence of human factors on the performance of software teams. This is consistent with the views of Larman and Vodde (2009) that ‘respect for people’ is a core pillar of LT and that leadership principles and ‘manager-educators’ are critical to its implementation. Poppendieck and Poppendieck (2003) presented seven principles of LT and proposed a mapping of software development practices to each of the principles. Hibbs et al. (2009) claimed that most writers on LSD have based their work on the principles espoused by the Poppendiecks. In recent years, LSD research has become more prevalent (Wang, 2011) and a number of international conferences have emerged.

In summary, LSD is included as an agile method by some and not mentioned by others. It is claimed that it should be seen as a management philosophy that applies LT to software
development methods and among other things, is useful to communicate strengths of agile methods to management. LSD is viewed as a strategic approach that may support the more tactical nature of ASD. Reinertsen, Middleton and others take a general view of LT in order to propose its judicious application in software development.

1.2.2 Agile Software Development

Avison and Fitzgerald (2003) stated that initial software development efforts were very chaotic and employed a very basic process of code and fix. This was untenable, and led to the creation and adoption of a step-by-step lifecycle based model. This initial predictive approach, published formally by Royce (1970), is known as the waterfall model. However, this approach was monolithic, creating a large time-lag for customers from initial requirements to acceptance of the completed software (Pfleeger and Atlee, 2005). Fitzgerald et al. (2002) summarised many approaches that emerged to counter these disadvantages, including prototyping, iterative, evolutionary and participative development.

All of these approaches (other than hacking) could be defined as planned or predictive, as even risk-driven spiral models have required milestones in order to maintain control (Boehm, 2002). The concept of rapid application development (RAD) changes the focus from predictability to adaptability, as this approach is more open to emerging requirements (Avison and Fitzgerald, 2003). This is a core differentiator of agile methods — they are adaptive rather than predictive.

Figure 1-4 presents how Boehm (2002) envisaged this range of approaches. The most basic approach is unplanned hacking. At the other extreme is the micro-planned contract-based approach, where all tasks have been decomposed to a very low level of granularity. The middle ground is addressed by planned iterative efforts. Agile methods and their associated adaptability come just before high-level milestone-based planned approaches.
Figure 1-4: Spectrum of Software Development Approaches (adapted from Boehm, 2002)

The origin of agile methods is related to the fact that many development approaches were too prescriptive and process-centric. As a result, practitioners developed methods to try and address these issues. For example, in his book ‘Rapid Application Development’, Martin (1991) used many of the insights gained from work performed by the DuPont organisation in their use of a development method called ‘rapid iterative production prototyping’. This approach to development was motivated by problems associated with SDLC-based development, and also by challenges surfaced by volatile requirements.

Rapid Application Development (RAD) aims to speed up development and lower costs without losing quality. Much research on RAD has been informed by practice, so it is difficult to find one firm definition or specification of RAD. Rather than focus on any one definition, researchers tend to look at the characteristics of RAD (Avison and Fitzgerald, 2006; Fitzgerald et al., 2002). These characteristics include: incremental development, time boxing, pareto principle, MoSCow rules, JAD workshops, prototyping, sponsor and champion and the use of specialized toolsets.

By the turn of the century, a number of ‘lightweight’ methods had emerged: Scrum, Crystal Clear, Adaptive Software Development, DSDM and eXtreme Programming. Characteristics of RAD are to be found in all of these approaches (Abrahamsson et al., 2003). In 2001, the 17 primary promoters of these methods met to discuss their work. They renamed this category of methods from ‘lightweight’ to ‘agile’ and issued a manifesto of their beliefs around software development (Fowler and Highsmith, 2001). Essentially the manifesto presents four values of agile methods:
• *Individuals and interactions over processes and tools*
• *Working software over comprehensive documentation*
• *Customer collaboration over contract negotiation*
• *Responding to change over following a plan*

*That is, while there is value in the items on the right, we value the items on the left more.*

(Fowler and Highsmith, 2001, para. 7)

The agile manifesto does not prescribe a particular set of techniques and guidelines — it is not a method in itself. The four values are supported by 12 principles that provide guidance on the philosophy of accepting the inevitability of change and dealing with this fact rather than planning for a predictability which often failed to be realised. These 12 principles are used in chapter 3 to investigate motivations for the application of ASD (specifically Scrum).

The creators of this manifesto did not deny the need for processes or tools — indeed they themselves had spent important parts of their careers designing methods, which is an indication of their commitment to disciplined development approaches. However, they felt that effective collaboration and communication between developers and other stakeholders was more critical than being adept at following tools and processes. This is consistent with the stages of team formation proposed by Tuckman (1965): *forming, storming,* and *norming* come before *performing* to full potential. Teams may become competent at following their processes and tools (norming) and never achieve the ‘performing’ stage where their activities are driven by comprehension of both their goal and their abilities to achieve it. Similarly, Fitzgerald et al. (2002) noted that novice developers initially tended to use methods as a crutch by strictly adhering to recommended tools and processes only to discard them as a hindrance to productivity when their skills increased. Mature developers developed the wisdom to use certain parts of a method that added value to their work (situational methods used ‘in-action’).
Fitzgerald et al. (2002) and Introna and Whitley (1997) indicated the dangers of teams placing too much focus on blindly following the prescribed guidelines of a method without appreciating the situation in which the method is being used. The first guiding principle of the agile manifesto recommends that developers collaborate effectively within the development context. This effective collaboration should result in developers understanding the reasons that they are performing a set of tasks and consequently motivate them to understand how their work relates to the collective goals of the project and team. The practice of ‘Metaphor’ within the eXtreme Programming agile method serves to support this concept (Beck and Andres, 2004). To follow Tuckman’s (1965) model, this comprehension should lead to the effective removal of aspects of the method that do not add value within the development situation, and ultimately lead to the team ‘performing’. The removal of aspects of a process that do not add value is a key principle of LT.

The agile manifesto promotes useful documentation. It seeks to emphasise that the software is the core of the development and it is critical that any documentation maintains its relevance to the most current version of the software. The development team should determine what documentation will add value to the core task at hand: the development of good working software. The third value accepts that while contracts form a necessary part of the structure of business transactions, it is important that customer collaboration is put to the fore in order to ensure that the development team elicit customer’s needs effectively and understand them completely. Contracts serve to set effective boundaries, but development details require closer collaboration.

Finally, the complexity and criticality of system requirements have been highlighted by many in the research community. As the system under construction evolves and comprehension of that potential system increases among both the customers and developers, new requirements emerge. The final principle of the manifesto notes that while it may be desirable to have planned a project, it is more important to have the
ability to react effectively when new unplanned aspects of the project emerge (Fowler and Highsmith, 2001).

In their systematic review of agile methods, Dyba and Dingsoyr (2008) identified the main agile methods as being: Crystal methodologies, Dynamic software development method (DSDM), Feature-driven development, LSD, Scrum and eXtreme Programming (XP). Abrahamsson et al. (2003) compared many of these methods and reported that the focus of Scrum is on the project management of software development initiatives. Many commentators present the assumption that Scrum is the most widely used agile method. However, despite indicators such as Scrum-master certification programs, claims of Scrum dominance by various consultancy practices and the prevalence of articles related to this method in many agile-related conference proceedings, little rigorous research has been presented to defend such claims. For example, Ambler (2007) expressed surprise that his survey on agile adoption revealed that despite their promotion by the Scrum community, ‘burn-down’ charts were not as widely adopted as he would have thought. Despite the lack of rigorous evidence, Dyba and Dingsoyr (2008) also assume the popularity of Scrum when they decry the lack of empirical research into its use. However, it appears that a critical mass of opinion exists that at the very least; Scrum is very popular among practitioners. Given that this ASD method is assumed to be very widely used and because it is a management approach, it was decided to focus the project management aspects of this study exclusively on Scrum.

1.2.3 Project Management of Agile methods

One of the largest not-for-profit advocates of project management globally is the Project Management Institute (PMI). This association has over 500,000 members. It publishes the guide to the Project Management Body of Knowledge - PMBOK every four years. This guide is in its 5th edition and is a recognized ANSI standard for project management. PMI (2008) presents the best practices of project management as a series of overlapping processes that can be grouped into one of five process areas (Initiating, Planning, Executing, Controlling and Closing). These processes are performed to meet the needs of
one of nine knowledge areas (Scope, Cost, Time and others). The structure of overlapping processes that feed into one another is based on the ‘plan-do-check-act’ (PDCA) model first introduced in quality management (Moen and Norman, 2006). Projects are characterised as once-off unique temporary endeavours to achieve a goal. The management of a project requires ‘progressive elaboration’ as project planning reveals more detailed information about the desired objective and work to be performed to achieve it. Traditionally, project management is viewed as a plan-driven concept where the project manager is the single point of accountability. This is considered a strength of the project management approach as it introduces predictability and a focus that can integrate and control tasks across different functions effectively. As stated in the previous section, agile methods fall into the adaptive side of the continuum of software development approaches. One might expect that project management would also need to extend its best practices to absorb the required adaptability of such methods. Project practitioners might argue that traditional project management practices address these concerns by their emphasis on strong change management and their underlying premise that project management processes are based upon the PDCA cycle. However, a counterpoint is that agile methods are ‘team-driven’ and as such not subject to the concept of a single point of accountability.

In their review of agile project management, Fernandez and Fernandez (2008) noted that performing agile project management may be about the judicious selection and application of different traditional processes. They summarized different management strategies that may be adopted based upon a project’s risk and uncertainty. They described proposals on how PMI (2008) may be used to support these strategies by expanding the concept of progressive elaboration and introducing an iterative approach to the execution of different process groups. It appears that there is scope for research in the field of adaptive project management. Currently, PMI (2008) presents iterative lifecycles as part of its best practices but does not present many more details on the best practices of adaptive or agile project management. However, in response to member needs, Agile tracks have been introduced in recent PMI conferences and an Agile community of
practice is in place among members worldwide. In 2011, an Agile project management accreditation was added to the PMI certification programme.

Cohen et al. (2004) discussed challenges of plan-based and agile methods, concluding that there is a need for both types of method depending upon the project environment. They noted that agile project management is most suited to rapidly changing environments where requirements are continuously changing and emerging. A further advantage of agile project management is its ability to detect risks early in the process.

As a follow-on to the agile manifesto, a group of project management researchers, authors and practitioners used their experiences with management of agile projects to propose a definition for agile project management, entitled “the declaration of interdependence” (Anderson, 2005). It is interesting to note the influence of LT on the statement of values produced by this initiative. Essentially, these values emphasize continual flow of value (including customer focus), strong emphasis on people, and flexible approaches to different project situations.

A similar approach to agile project management is adopted by Rico (2010) in his explanation of the motivation for this methodology. He stated that agile project management should not be interpreted as the implementation of some subset of project management practices but rather as a new project management ‘paradigm’. He used (PMI, 2008) to explain that traditional plan-based project management is based upon many of the management science theories that emerged in manufacturing and as a result is deductive in nature. This is evident by the reliance of traditional project management on the core ‘Work Breakdown Structure’ (WBS) construct to act as a foundation for much of the planning and subsequent execution and control of projects. This construct is a deliverable-oriented hierarchical decomposition of all the work to be done in the project in order to achieve its goal. The ‘project scope’ defined in a WBS is a key input to most detailed project planning activities including scheduling, budgeting, risk management and communications planning. An incomplete WBS will lead to the need for change
management and potential rework during project execution. Agile project management emerged as a mechanism to deal with unstable environments and is based upon complex adaptive systems theory - it is exploratory or inductive in nature. Koskela and Howell (2002) asserted that traditional or PMI defined project management is based upon a transformational view of production and that it does not address the operations theories of flow and value (which are the foundation of much LT). Given the complex nature of software development and the difficulty in managing stakeholders’ needs (Brooks, 1987), there appears to be a strong argument for an inductive approach to such projects.

However, the premise that the best practices presented by (PMI, 2008) are not relevant to ASD project management is questionable. It is clear that the WBS relies on a certain aspect of deductive reasoning. However, as stated earlier, PMI (2008) describes ‘progressive elaboration’ as a core characteristic of projects. It also provides for iterative structures as an option for project lifecycles and crucially, WBS construction includes a technique called ‘Rolling Wave planning’. This technique enables the WBS to evolve in later iterations of the project as more information emerges. Therefore, this approach can be used to support exploratory research by a combination of deductive-inductive reasoning as the project progresses and converges upon the goal of meeting all stakeholders’ requirements. Furthermore, the PMI propose that best practices of project management reflect the implementation of the adaptive ‘plan-do-check-act’ (PDCA) model framed by initiating and closing processes. The PDCA model is applied via particular planning, execution and controlling processes. This is consistent with the presentation of the SCRUM agile method by (Schwaber, 1995). He stated that the SCRUM lifecycle contains two theoretical or predictable phases: initial planning and release. These two phases form the endpoints of an empirical or adaptive series of planning, execution and control processes named ‘pre-sprint’, ‘sprint’ and ‘post-sprint’ activities.

Rico (2010) presented a number of agile project management methods including Scrum and XP project management. Initially created by Jeff Sutherland, Scrum was formally
introduced to the software development world at OOPSLA ‘95 (Schwaber, 1995). Inspired by the work of (Takeuchi and Nonaka, 1986), Scrum was built from the best practices in new product development. The essence of Scrum is to nurture evolution of a software system by facilitating adaptability to changing circumstances as the development team learn more of the domain and system being created. The method aims to have a team focus on the most important work to be done and to enable them to work in an integrated coordinated way to perform this work effectively. It is a high-level management framework that does not prescribe particular practices. In order to simulate the flow of work, Scrum has multiple iterations or ‘sprints’. A set of requirements contained in a ‘product backlog’ represent the needs of the customer. At each sprint, the team determines what features from the backlog to produce and they place these in a Sprint backlog. Daily meetings and clear understanding of both outstanding and completed tasks within the sprint (iteration) maintain team momentum and focus. On completion of the sprint, the team reviews the work in preparation for future sprints and to implement any process improvements deemed useful (Whitaker, 2009; Cohen et al., 2004; Sutherland, 2008). Scrum embraces the operations concepts of flow and value rather than the transformation concept addressed by traditional project management (Koskela and Howell, 2002).

Many commentators propose that eXtreme Programming (XP) is the most widely used agile method. Fowler (2001) claimed that its relative success is most likely due to its applicability. It provides realistic development techniques that align the work with the principles of the agile movement. Created by Kent Beck, XP is a method that was built from industrial experiences. XP presents a set of key practices and recommends that a team select the practices they determine to be most appropriate to their development situation. Fitzgerald et al. (2002) focussed on XP because it is truly a ‘method-in-action’. In one of the earliest publications to describe XP, Beck (1999) acknowledged the influence of Scrum on the creation of XP.
Abrahamsson et al. (2003) described the lifecycle of an XP project as having six phases: ‘Exploration’, ‘Planning’, ‘Iterations to Release’, ‘Productionization’, ‘Maintenance’ and ‘Death’. Exploration involves working with the customer to build a series of ‘stories’ that drive the releases. The team plans the short iterations that make up a release cycle. They then execute these iterations and put the completed release into production. This release is then maintained while the next release cycle begins its process by verifying the outstanding list of stories and delivering them. Death occurs when a system that allows the customer to perform all of their stories has been delivered.

Ceschi et al. (2005) compared project management issues encountered by plan-based and agile projects. A notable finding from their work was that despite a general awareness of agile among many plan-based managers, the main barriers to agile adoption was lack of in-depth knowledge on agile methods and the difficulties of agile application among geographically dispersed teams. This dissertation intends to contribute to the body of knowledge in both of these particular areas.

While acknowledging that much work is required in the field of agile project management, it is not the intention of this study to investigate this particular concern. A simplistic approach is taken that SCRUM is a popular project management framework used to manage agile projects and that this method can be used as a vehicle to explore this aspect of ASD.

1.2.4 Global Software Development

GSD is considered to be a subset of distributed development in that the team members are distributed globally (Lings et al., 2007). There are many configurations of distributed development teams (Grinter et al., 1999; Lamersdorf et al., 2009). Examples range from remote sub-teams producing specific modules of a product to teams where different functional roles such as programming or business analysis are executed in different locations.
Teams that conduct GSD may gain certain advantages over those that are co-located. Conversely, distributed teams may encounter obstacles that are not major issues in co-located teams. In their review of published literature on distributed development, Ågerfalk et al. (2005) proposed a framework of the benefits and challenges related to this field. Although this framework is built from the broader subject of distributed development (including open source development), many of the issues identified by the framework relate to globally distributed teams. This framework has been leveraged in various GSD studies (Holmstrom et al., 2006; Lane and Ågerfalk, 2008; O’Conchuir et al. 2009). It has also been refined to describe both GSD and intra-national distributed development (Lings et al., 2007). As such, it is deemed appropriate as a general mechanism to support the GSD aspect of the research.

1.2.5 Background - summary.

LSD is useful as a means to communicate effectively with both management and developers. It serves to align the tactical nature of agile methods with an organization’s strategy. The principles of LT have influenced different researchers to comment on LSD. ASD is an umbrella term for adaptive methods that emerged to counter difficulties with traditional predictive software approaches. These methods follow the values of the agile manifesto. Project management is associated with predictive planning and control, but is increasingly applied to the planning and control of agile methods. Scrum, an agile management method, is arguably the most widely used agile method. Both Scrum and LSD emerged to address issues with complex adaptive systems. GSD is a very popular approach to software development, motivated by opportunities to decrease costs and increase availability of time and access to diverse skill sets.
1.3 Objective and research questions

1.3.1 Research objective and key contributions

The objective of this thesis is to present a deep understanding of the relationship between LSD and ASD in order to address the confusion noted in section 1.1. This broad objective is reduced to a more detailed goal in order to establish a clear project scope. The detailed goal serves to focus the research and also tackle the three research gaps identified in section 1.1:

(i) Address the confusion between LSD and ASD
(ii) Provide a rich description of project management within ASD
(iii) Examine the application of ASD in a globally distributed environment.

This detailed goal is to conduct empirical investigations into the relationship between LSD values and a particular application of an ASD management method (Scrum). In order to perform this work, an aspect of this goal is to propose a model that will support these investigations. It is not the intention of this study to validate the model. It is intended to be a conceptual framework that supports research activities. More specifically:

“Develop a model and use it to support an investigation of how SCRUM may leverage LSD values when used to manage projects that are performed by distributed development teams.”

This study provides three research contributions:

1. A research framework. This is a description of the mechanisms used to guide and support research activities. It includes detailed records on literature analysis and a case study protocol which may be used and replicated in future research.

2. A conceptual model. This model applies the method rationale analysis framework to link LSD and ASD, so that some clarity may be applied to the confusion between these two software development paradigms.
3. Research into the application of ASD within globally distributed settings. As per the observations of gaps in the agile research space (Abrahamsson et al., 2009; Ågerfalk et al., 2009), this study conducts an empirical investigation into the application of ASD in a globally distributed setting.

A further practical contribution is a set of recommendations for the organization studied on how to achieve more value from their application of distributed SCRUM.

1.3.2 Research Approach and questions

This study analyzed both scientific (peer-reviewed publications) and practitioner (books, guides, web logs and guides, textbooks) literature in order to propose globally distributed Scrum practices and goals and LSD values. The method rationale analysis framework (Ågerfalk, 2006) was used to develop a conceptual model associating LSD values with the goals (or principles) and practices of distributed Scrum development. This model facilitated empirical research in order to induce knowledge on any LSD values realized by globally distributed Scrum projects. Each lean value induced was associated with particular globally distributed Scrum practices and associated goals (or principles).

The purpose of the study is to consider the following question:

“What LSD values are present in the performance of activities within projects conducted by globally distributed teams using the Scrum framework?”

The method rationale analysis framework leveraged by Ågerfalk & Fitzgerald (2006) was used in this study to form the linkage between LSD and ASD (more specifically, Scrum). The framework enables the association of LSD values to the goals (or principles) of distributed Scrum which in turn are linked to the practices of that method. Background literature reviews on LSD, Scrum and the application of Scrum in globally distributed settings were conducted in order to derive the required LSD values, distributed Scrum principles and their associated practices.
A structured approach was adopted to the review of LSD literature (Webster and Watson, 2002). Two major contributions to lean literature, Ohno (1988) and Womack & Jones (2003) were used as the basis for a series of backward and forward citation searches in order to establish a core set of sources. In keeping with the recommendations of Webster and Watson (2002), sources were then organized into different themes by using MS-Excel to develop a ‘concept matrix’. Four concepts were identified: ‘operations management’, ‘product development’, ‘project management’ and ‘software development’. Sources from each concept were further described using lean principles identified within each paper as units of analysis. As recommended by Levy and Ellis (2006), the identified lean principles from the various sources were then synthesized into a set of LSD values representing LT literature.

The literature review of globally distributed Scrum principles pursued a targeted approach. Three sources were used to propose a set of Scrum principles or goals. The first paper by Takeuchi and Nonaka (1986) was used as it is cited as the inspiration for the creation of the Scrum method (cf. section 3.2.1). Fowler and Highsmith (2001) is reviewed as it presents twelve core principles associated with ASD (cf. section 3.2.2). Finally, GSD challenge alleviation goals are proposed from a peer-reviewed literature review on distributed development conducted by Ågerfalk et al. (2005).

Twelve core Scrum practices were identified from a review of three specific guides that have described Scrum from its initial foundation through to 2011. An additional review of recently published practitioner literature was conducted in order to identify many supplementary Scrum practices reported in both regular and globally distributed settings. While not exhaustive, these additional practices served to enrich the analysis of how the identified twelve core Scrum practices may be associated with the specific NPD, ASD and GSD principles.
The method rationale analysis framework was then applied to these findings in order to develop a candidate Lean Distributed Scrum model (LDScrum). Figure 1-5 illustrates a template of the model.

![Figure 1-5: Application of Method Rationale Analysis Framework to LSD and Distributed Scrum](image)

Development of the initial version of the LDScrum model was the first step in a two-step research approach (figure 1-6). The LDScrum model formed the basis for data collection and analysis activities. A case study was conducted in order to uncover rich descriptions related to distributed Scrum activities. These idiographic details were then applied to the initial value-set. The synthesized findings from this research resulted in a refined version of the LDScrum model.

![Figure 1-6: High-level research approach](image)
1.3.3 **Study Work Breakdown Structure**

Section 1.2.3 described the work breakdown structure (WBS) as a core project management construct that defines the project scope and forms the basis for detailed planning and execution of project activities. Figure 1-7 presents the WBS for this research study. All the work required to produce both the candidate and refined LDScrum models are detailed in this artefact. The work segments presented in the second layer of the WBS (1.1, 1.2, 1.3, etc.) equate to the contents of different chapters. The high-level overview of the thesis structure presented in the next section offers a little more detail on each chapter. Throughout this thesis, each chapter commences with a diagram presenting the segment of the WBS to be addressed by that chapter. This is intended to position each chapter in the context of the overall project scope.
Figure 1-7: Work Breakdown Structure of research study
1.4 Thesis structure

Chapter 2 identifies a set of LSD values and the associated benefits to be realized through the achievement of these values. A genealogical approach is taken here to investigate the origins of LSD and determine the various principles that have been reported by different commentators as ‘LT’ has emerged from manufacturing to its application in product management, project management and software development. The resultant set of principles is then used to derive a candidate set of lean values.

Chapter 3 presents a background on the SCRUM agile management framework. Certain characteristics of successful product development environments are combined with the 12 principles of agile software development to form a set of goals that may motivate particular Scrum practices. A core set of Scrum practices are identified and a targeted literature review induces many supplementary Scrum practices associated with each core practice. The chapter concludes with a table outlining the association between each core Scrum practice and various goals.

Chapter 4 extends the work of the previous chapter by applying a similar analysis to the application of Scrum in GSD. Particular goals related to overcoming published challenges of GSD are declared. A targeted literature review reveals various supplementary practices performed in distributed teams using Scrum. This analysis results in a table presenting each core Scrum practice with the particular GSD goals that may motivate its performance.

Chapter 5 presents the candidate LDScrum model which is used to inform the empirical research. The theory that underpins construction of this model is explained. The manner in which the outputs of chapters 2, 3 and 4 are combined with method rationale analysis theory to develop this model is described.

Chapter 6 details the research methods leveraged for this study. It presents the researcher’s ontological and epistemological position and explains how this position
informed selection of the interpretive case study method as the preferred approach used to conduct empirical research.

Chapter 7 describes the research design process. It explains the motivations for different approaches taken in reviewing literature related to the concepts under research. In order to provide benefit to future researchers who may wish to replicate aspects of this study, detailed design decisions are explained and a clear description of the case study context is provided.

Chapter 8 presents a detailed analysis of the empirical research findings. This results in a rich description of the findings associated with an ISD project performed using Scrum by a distributed ISD team that is configured in a ‘Scrum of Scrums’ structure.

Chapter 9 extends the empirical findings presented in the previous chapter. It describes another ISD project performed within this case. This particular project is performed by a distributed ISD team performing Scrum using the ‘Totally Integrated Scrum’ team configuration.

Chapter 10 presents the refined LDScrum model. Insights gleaned from the research are highlighted. The main contributions of this research are presented. Limitations of the study are outlined and future research opportunities, recommendations and directions arising from this study are detailed.
Chapter 2- Lean Software Development

2.1 Introduction

Figure 2-1 below highlights the work addressed in this chapter. These work packages form part of the overall project scope provided in the WBS (figure 1-7).

Figure 2-1: LSD values WBS segment

The purpose of this chapter is to induce a set of higher-order values that underpin lean principles and practices. Section 2.2. presents an exploration of the origins of lean concepts and their application in different domains. This work identifies a series of principles reported in literature on the lean paradigm. In order to support the treatment of LT within software development, section 2.3 presents a table relating lean terms and concepts to the application of ISD activities. Finally, the chapter concludes by reporting on the results of a hermeneutic interpretive process used to synthesize a compilation of the aforementioned lean principles into a set of lean values. These lean values are used in later aspects of this research as LSD values that may be leveraged to relate LSD to ASD.
2.2 Evolution of Lean Software Development

Methodology (or the study of methods) has proposed that software methods can be viewed as formalized approaches created in laboratory environments. In many cases, these approaches are subsequently tailored to become both situational methods to suit certain contexts and methods-in-action adapted as a result of the influence of various forces in a particular development situation. Many methods have been based upon a higher-order construct of problem-solving known as the waterfall development lifecycle (Fitzgerald et al., 2002).

It is proposed in this thesis that LSD should also be viewed as a broad concept that informs the construction, adaptation and application of many methods: a management philosophy that has evolved from different domains including operations management, product development, project management and software development. This section reviews research into the application of LT in these domains.

It is evident from the progression of various lean manifestations (figure 2-2) that contributions to this philosophy have been made by both formalized methods and methods-in-action. This chapter addresses the origins of LSD, describing reported opinions of selected commentators from each domain. A selection of principles are identified and presented in the context that they are proposed by the original source. To supplement this narrative and support further inquiry, each source is mapped to a codified list of its associated principles (table 2-1). Each principle referred to in the following sections is supported by its associated table 2-1 code e.g. On refers to principle n of (Ohno, 1988); SBn refers to (Spear and Bowen, 1999). The chapter concludes by proposing a set of LSD values that are induced from a synthesis of the LSD principles. These principles and values are then used in subsequent chapters to support empirical research on the relationship between LSD and Scrum development in distributed teams.

2.2.1 Influence of operations management on LSD

The Toyota Production System (TPS) is often cited as one of the key catalysts to the worldwide interest in lean approaches to different disciplines. Many TPS concepts
have been practiced by manufacturers worldwide (Spear and Bowen, 1999). This system evolved within Toyota’s manufacturing environment in response to the challenges facing the company to be cost-effective and adaptive in order to initiate and sustain a competitive advantage in the automotive sector (Hibbs et al., 2009; Coplien and Bjornvig, 2010; Ohno, 1988; Holweg, 2007).

Many early developments such as parts interchangeability, work standardization and division of labour led to the industrial revolution in which craft workers were replaced by mass production. The US automotive industry was revolutionized by these concepts as Ford employed them to great effect in the mass production of the Model T.
motor car. Further innovations such as model variety were introduced by General Motors. These advances enabled the US automotive industry to dominate the sector world-wide. By 1945, the Toyota Motor company launched a programme to ‘catch up with America’. The Japanese automaker had severe constraints including a small home market, limited land availability, skills shortages and limited funds for capital investment. These constraints necessitated the production of “small quantities of many varieties under conditions of low-demand” (Ohno, 1988). This requirement motivated Toyota to apply innovative manufacturing techniques in order to find an alternative to established mass production approaches (Ohno, 1988).

Figure 2-3 presents the various components that formed the foundations of the Toyota Production Systems (TPS). Proven operations management techniques were applied e.g. Taylor’s empirical methods to define standardized work. Quality concepts including statistical process control were leveraged as were analyses of innovations adopted by other industry sectors (such as supermarket systems). A culture of continuous learning and work improvement was promoted through the introduction of various initiatives intended to reduce waste and increase value to the customer.

Figure 2-3: Evolution of Toyota Production System (TPS) (UL, 2005)
Ohno (1988) challenged the use of large batch production in the pursuit of economies of scale by stating that large batches both increase waste (such as defects) and inhibit product diversity (Holweg, 2007). Waste reduction (O1) is a key principle of the TPS. He classified ‘muda’ or waste into seven categories: Transportation, Inventory, Motion, Waiting, Defects/rework, Overproduction and Over-processing. Levelling customer demand (O6) is key to enabling the transformational flow of parts through the supply chain. The Just-in-time (JIT) method is used to enable flow (O2). This mandates that a part is produced at the time it is needed. It requires both upstream activities to have rapid set-up systems (in order to respond to demand) and downstream activities to have levelled their schedules (in order to enable predictable flow). Ohno (1988) noted that the arrival of just enough inventory just in time enables waste reduction, promotes consistency and reduces the chances of employees being overburdened by workloads. For these reasons, the principle of flow is emphasised by many authors. Built-in-quality (O3) refers to the concepts of mistake-proofing in order to prevent the occurrence of defects. A related principle specifically identified in the work of Ohno (1988) is ‘autonomation’ (O7) which refers to quality promotion through the use of ‘intelligent machines’. Such machines (or systems) are capable of suspending work and alerting operators upon detection of adverse quality events. An example of a software development practice that would follow the principle of autonomation would be a build verification test that reacts to any test failures by automatically stopping the build and waiting for developer intervention. Cost reduction (O9) demands that a product or service selling price is dictated by the market - not on a cost plus basis. This serves as an incentive to drive down cost. Other principles highlighted by Ohno include the empowerment of workers (O4), awareness that high-growth success criteria (such as mass production) do not apply in slow-growth economies (O5) and ensure all improvement initiatives should be a response to a clear opportunity or need (O8). Spear and Bowen (1999) investigated the difficulties of TPS replication encountered by organizations worldwide, despite Toyota’s willingness to offer open access to their manufacturing systems. They postulated that this difficulty emanates from imitators placing an inappropriate focus on the practises of the TPS and not enough emphasis on the overall implementation of the system. They proposed that the implementation of TPS can be defined as the compliance to four rules: detailed specification of every task (SB1); unambiguous
communication procedures between employees (SB2); a single pathway for every product or service (SB3); strict adherence to the use of the scientific method in pursuing improvements (SB4). This leads to an organization equipped to pursue continuous learning. The TPS promotes the encapsulation of work so that modifications to an area do not lead to unintended adverse consequences in other parts of the system. Liker and Morgan (2006) noted that this continual pursuit of innovation at the ‘Gemba’ or place of action had a consequence that the overall system was not formally documented although ultimately, over time, a broad view was introduced as the Toyota Production System House (figure 2-4). Details on the various terms outlined in this figure are presented in section 2.3.

![The Toyota Production System House](image)

**Figure 2-4: The Toyota Production System House (Liker and Morgan, 2006)**

Many studies had been conducted of lean operations following publication of the initial Japanese version of Ohno (1988) in 1978. However, it was the publication of Womack et al. (1990) combined with the establishment of various subsidiary car plants in the USA that led to the dissemination of lean operations concepts in western manufacturing (Holweg, 2007). Womack and Jones (2003) described various experiences of TPS lean initiatives in many different environments in western organizations. This work challenged the notion of full utilization of expensive assets by batch processing operations and stated that process reengineering is not sufficient
to transform organizations. It recommended the combination of both high-level and granular approaches to foster operational improvements:

- Exceed a process viewpoint by conducting a low-level identification of all activities that add value to the customer
- Expand beyond the boundaries of the organization to consider the entire supply chain and its influence on the effective performance of work.

Womack and Jones (2003) synopsized their interpretation of the TPS and additional lean research into five principles: value, value stream, flow, pull and perfection. In an ISD context, Value (W1) may refer to customer acceptance and approval of an application feature. This is synonymous with the concept of ‘big Q’ quality espoused by Kan et al. (1994). Value stream (W2) refers to the mapping of every step from initial transformation of raw materials to delivery of final product in order to identify and tackle value and waste. In ISD, steps in a software method could be analysed to identify ‘value added’, ‘non value-added’ and ‘necessary non-value added’ activities being performed in a software project. Flow (W3) is concerned with the establishment of a process of value-adding activities resulting in the delivery of a product or value. Flow in operations management promotes the reduction of batch sizes in order to have a product flow through production out to a customer. In an ISD context, the concept of a batch of products is equivalent to a set of application features. The difficulty of establishing a ‘standard feature’ is sometimes addressed by the notion of feature sets or ‘minimum marketable features’ (MMF) to deliver value (Middleton et al., 2005; Middleton and Joyce, 2012). Pull (W4) refers to the establishment of production to react to customer demand. The process begins with the final customer being synchronized with product completion. No upstream work is delivered until the immediate succeeding downstream operation signals a need (via Kanban) (Ohno, 1988). The Scrum framework may be leveraged to apply pull in an ISD project (Hibbs et al., 2009). Perfection (W5) relates to the continual improvement of standard work.

This section presented principles uncovered from literature related to the application of LT in the manufacturing domain. The application of LT within the Toyota manufacturing organization and the evolution of the TPS have had a major influence on this field. A timeline presenting the introduction of different work practices
Throughout the evolution of the TPS can be found in (Ohno, 1988). Many other studies and observations on the application of LT within manufacturing have been performed and one of the most notable of these has been the work of (Womack and Jones, 2003). An overview of the major events and publications between 1932 and 2004 that led to the dissemination of lean production is provided in (Holweg, 2007).

### 2.2.2 Influence of Product Development on LSD

Liker and Morgan (2006) explained that lean principles extend beyond the manufacturing production department. Many different industries have adopted different techniques from the TPS practices. They proposed that lean adoption in a non-repetitive environment such as services would be better served by using the experiences of the Toyota Product Development system (TPPD). Studies of this system were synthesized into a generic lean product development framework of management principles that address three areas:

(i) **Process**: process thinking promotes a complete understanding of what value is from the customer’s perspective (L1), strong up-front design to ensure the correct plan is followed (L2), effective levelling of planned activities (L3) and establishment and adherence to standard processes (L4).

(ii) **People**: the management of this area fosters a culture of accountability, competence, empowerment and technical and domain expertise (L5, L6, L7) that enables workers (including suppliers) (L8) to continually improve both the product and the process. An overarching characteristic of the TPPD is the promotion of continual learning by the workforce (L9). The application of certain lean techniques in order to remove some waste from a particular service or job is secondary to an organization gaining momentum in the empowerment of its workforce to constantly pursue continual improvement (L10) and evolve new standard activities that form the basis for future waste reduction initiatives.

(iii) **Tools/technology**: tools should be tailorable (L11). In an ISD context, customization of tools would be recommended to ensure that a tool or technology does not become the focus of the developer rather than the information system they are tasked with creating. For example, commercial...
development frameworks incorporating complex integrated development environments (IDE) may require many months of training before a developer is competent using them. Such environments should be customized to ensure they are relevant to the development situation. Also, tools and technology should be used to support communications (L12), standardization and continual learning within the organization (L13). Communication is enabled through simple tools and methods. Liker and Morgan (2006) explained how the Hoshin Planning method of objectives alignment is used to link high-level product (vehicle) objectives to specific sub-system goals and that clear visual communication tools are leveraged to support application of the method. Liker and Morgan (2006) explained that in Toyota, continual improvement through ‘kaizen’ events is not possible without standard work approaches in place. They referred to engineering checklists managed at the technical component level by engineers using tools to evolve ‘local’ standards. Configuration management tools may contain standards to ensure code quality prior to acceptance in a product build. Such tools may incorporate processes to enable developers to augment code quality standards as their knowledge grows within their development context.

Liker’s interpretation of TPPD is consistent with recommendations on the application of total quality management to software development: innovative approaches to customer care, continual process improvement, judicious use of tools and technologies and strong awareness of human factors (Kan et al., 1994).

Reinertsen and Shaeffer (2005) also emphasized the difference between product development and manufacturing activities. Process variability, sequence, volatile requirements and attitudes to risk influence the manner in which lean principles are addressed in a research and development (R&D) environment. Whereas variability in a repetitive environment is to be avoided, this is not the case in a non-repetitive situation. In R&D, variability is desirable in certain circumstances and is to be removed in others (R2). Manufacturing adds value to items in a sequential manner. R&D adds value to an intangible asset: information. As such, this value addition can occur to the same information in different places at the same time, thus necessitating
consideration on how the potential parallel addition of value can be exploited. R&D deals with moving targets as market needs and technology capabilities evolve. It must embrace uncertainty (R6) in order to innovate whereas manufacturing aims to enhance productivity by operating under stable constraints. However, despite the aforementioned differences between manufacturing and R&D, they also proposed that certain lean principles discovered by the manufacturing sector could be applied to product development. Reduction in batch sizes equates to addressing smaller ‘batches’ of information (R1). This can be interpreted as the reduction of project size (duration or budget). Variability of expected work and task durations are an outcome of risk-taking which is critical to innovation (R8). Enabling workers to be multi-functional (R7) promotes a ‘pull’ strategy to deal with task variability and also encourages the maintenance of flow (R3). This approach runs contrary to a predictive planning approach which can often be foiled by latent requirements or emergent project paths.

The main difference between lean product development (LPD) and traditional product development is the focus on customer value provision at each process stage. LPD embraces long-term thinking and encourages sustainable development processes that enable an organization to manage changing customer needs and market conditions. However, such processes are not well defined in the literature. As a result, organizations must determine appropriate processes that address LPD concerns (von Wurtemberg et al., 2011).

### 2.2.3 Influence of Project Management on LSD

Project management consists of a series of overlapping processes or activities that address the needs of nine knowledge areas. One of these knowledge areas is risk management and its processes include risk identification, analysis, response creation and control (PMI, 2008). Koskela and Howell (2002) asserted that a project management theory that espouses the up-front removal of uncertainty is based upon the production theory of transformation and does not address other production theories such as flow and value. Charette (1996) contended that risk management should be considered a central tenet of software project management rather than just
one of a number of different areas to be managed. He based this proposal on the proposition that the complexity of software systems made them unsuitable for the traditional or scientific problem-solving approach to their resolution. The dearth of reliable information in software projects makes them notoriously difficult to predict and this has led to many notable project overruns. Risk management has been the vehicle used to try and improve software project management control. The Spiral model presented by Boehm (1988) aims to encapsulate risk management activities in the project lifecycle in order to enhance project control. Charette maintained that for large ISD projects, the risk management paradigm should be extended to all levels of the project team and consistent defined decision making processes should be established to support it. This is necessary as such projects often involve multiple stakeholders with differing expectations, uncertain information and high-impact consequences of failure. He described this as the ‘joint approach’ leading to both top-down and bottom-up views of risk in order to incorporate both the high-level strategic external aspects and the detailed technical aspects of an opportunity or threat. This required strong leadership and an awareness of the relationship between an information system under construction and the environment in which it is to be used.

Charette (2003) claimed that software engineering research has been stymied by the mapping of the engineering approach to the project paradigm. Different project-improvement processes have been championed such as the systems development lifecycle (Royce, 1970), prototyping and iterative approaches. Process improvement research has attempted to tackle the difficulties of this mapping by expanding the focus from individual projects to organization capability. This has resulted in organizational maturity models such as CMM and ISO-15504 (Zahran, 1998).

It seems reasonable to accept that project and process improvement initiatives have had some positive influences on software engineering. However, Charette (2003) stated that software engineering is yet to address sufficiently the challenges of change, complexity and uncertainty. The agile movement have decided that rather than trying to control these challenges, it is preferable to address them by engaging with the user and building ‘simple complex systems’ that allow rapid feedback and unforeseen functions to emerge. The pursuit of this approach leads to self-reinforcement which in
turn results in enhanced team performance. Ultimately, this can expand beyond the
software development group and lead to a link between IT and strategic business
thinking, thus supporting the adaptive enterprise (Charette, 2003).

The Lean Development (LD) ‘agile method’ supports the application of this approach
(Charette, 2003; Highsmith, 2002). LD is a top-down strategic approach that proposes
a management philosophy rather than practices and techniques. This approach
emerged from Charette’s views on risk management and organizational maturity. The
approach leverages risk as an asset that may be used by an organization to provide
value to customers. Just as Kichoro Toyoda set the ‘impossible goal’ to his workforce
to catch up with the US automotive giants in three years, Charette (2003) set a goal
that LD should enable teams to establish and sustain massive productivity gains. He
proposed that LD projects should be completed using 66% less time and cost and
produce 66% less defects than if they were performed by a CMM level 3 certified
software development organization using other approaches. This is consistent with the
view that in order to make a successful and sustainable transition to lean, it is
important to provoke ‘kaikaku’ or radical improvement by setting extremely
challenging goals (Middleton and Sutton, 2005). LD supports the extension of
information technology work into the business, thus increasing the ability to provide
value to the customer (C1). It is a tactic employed to evolve a ‘change tolerant’
organization (C4) that is capable of predicting the needs of the market. It requires
collaboration between customers, marketing, business management, project
management and software developers (C2, C3).

LD involves a two phase process encompassing business and technical considerations.
A meta-process is performed by domain-specification teams that address the
identification and delivery mechanisms for customer value. Design-build teams
follow a micro-process that governs the software development using one of a portfolio
of processes. It is recommended that an up-front architectural approach is used to
develop a domain-specific ‘expressive system’. Such a system allows users to
construct solutions for their needs and ultimately promote the development of change-
tolerant software i.e. software systems that contain emergent properties not fully
envisaged at the time of their construction (e.g. spreadsheet).
LD may be summarized as a method that evokes a strategic focus, encompassing the concept of risk entrepreneurship, leveraging the principles and lessons of lean production and employing stretch goals to motivate its adoption (Highsmith, 2002).

As stated in the introductory section 1.2.3, a definition of agile project management was published in 2005 with the intent of promoting debate and discussion on this topic among the ISD community (Anderson, 2005). It is notable that many of the values presented in this definition reflect the lean principles discussed in these sections.

2.2.4 Influence of Software Development on LSD

During the 1990’s, various reports were published on different initiatives that applied lean concepts to software development approaches. Hou (1995) evaluated different approaches in order to propose a lean hardware/software method to be applied to the development of embedded systems used by the U.S. military. Different criteria were proposed and used to evaluate a series of existing development approaches. These criteria were generated from process research, lean product development concepts and hardware systems design considerations. The final set of 14 evaluation criteria reflected concerns related to process characteristics and system design issues. Although, these systems design considerations focussed on embedded or hardware/software development issues, this thesis reflected on these evaluation criteria from a general ISD perspective (Hou, 1995).

(i) Process characteristics: The method to be used must be clearly defined (H1) and be sufficiently transparent (H4) so that it may be effectively controlled. Another process characteristic that supports the concept of control is continuous improvement (H9) as feedback systems enable the method to evolve. This is consistent with key tenets of process thinking (Zahran, 1999). Effective definition should support customization of the method (H5). It also should be scalable (H7) to meet different development needs. It must leverage configuration management (H2) activities in order to ensure effective change control. User engagement (H3) is vital as is rapid development cycles (H6) to
ensure currency and relevance of the system being developed. Defect-prevention (H8) supports the development of high-quality systems as it is more effective than testing. The removal of waste appears to be synonymous with the concept of lean and the presence of defects is one of the more visible elements of waste in a software application. Cleanroom engineering places a strong emphasis on defect prevention, focusing on the root cause of a defect’s presence in a system and promoting the correction of the development process so that such a cause is removed into the future.

(ii) Systems design considerations: The system should exhibit ‘manufacturability’ (H10). In an ISD context there needs to be effective system deployment mechanisms. Both corrective (H11) and perfective (H12, H13) maintenance concerns described by Lientz and Swanson (1978) need to influence system design. Finally, design activities should embrace hardware and software dependencies. One aspect of this consideration is addressed by the area of performance engineering.

Middleton (2001a) emphasized the importance of defect prevention (M3) in his explanation of how lean concepts introduced by (Womack and Jones, 2003) relate to software development. He explained that system specifications may be considered work-in-progress and developer misunderstandings of requirements constitute a risk of waste. In software development this waste can manifest itself in a defect that could have been quickly resolved during analysis and specification creation but could require much greater effort if uncovered at a later stage in development. The reason for this is because an analyst’s understanding of a functional area and consequences of issues around that area are much deeper when the analysis and specification of the area are their primary work task. Furthermore, at a later development stage, although the defect may be resolved, no effort may be taken to address the reason for its introduction via an invalid specification. So the potential for a reoccurrence of a similar issue is not addressed. He described reports of defect prevention practices using checklists and root cause analysis. He suggested that the lean approach to process is effective once a systematic process is in place. This view is consistent with (Charette, 2003) that lean principles are a progression from the project management and process improvement movements. Middleton (2001a) reported five principles of
LM and proposed that these can be applied to the software development process. In an earlier work, Middleton (1995) referred to one of these principles, ‘defect prevention’ (M3) as a key principle of just-in-time (JIT). The principle of continuous improvement (M1) is enabled by worker empowerment (M2), clear visual quality measures (M4) and automatic quality measurement mechanisms (M5). The use of participatory action research to investigate the application of these principles to software development revealed team and HR issues blocking sustainability of the lean practices. Although the timeline of this experiment was very short, a more extensive study of LSD application is reported in his later work.

Middleton et al. (2005) described a two year investigation of how various lean principles were leveraged in order to resolve a systemic problem of ongoing delays caused by defect resolution following code completion. Many product development issues were evident in this organization. Principles applied by the team included: Continuous-flow processing (small iterations of fully tested functionality) (MF1); Customer-defined value (intense emphasis on requirements elicitation) (MF2); Design structure matrix to enable flow (estimation and chunking of requirements using function point analysis to support resource allocation to work) (MF3); Takt time (Grouping of requirements into ‘units of work’ so that a takt time could be calculated and used to assess team productivity rate) (MF4); Linked processes (reduce distance between collaborators and related processes) (MF5); Standardised procedures (consistency of roles, names, work practices etc. to facilitate resource allocation between projects) (MF6); Eliminate rework (defect prevention using root-cause analysis (MF7); Scope management enhanced by increased focus on customer needs and context) (MF7); Posting results (feedback on productivity and errors thus promoting continual learning) (M4, MF8); Data driven decisions (impartial data collection thus avoiding delays due to meetings and disagreements) (MF9); Minimize inventory (decompose product components into stories of 3-5 features of 3-5 units of work to be tackled separately by multi-disciplinary teams (developers/QA/Marketing)) (MF10). Following the application of these principles to ISD over a two-year period, a review of the organization revealed significant improvements. Key findings included:

- 25% productivity gains
- Strong staff support for lean concepts
- Substantial increase in product quality
- Major decrease in defect resolution effort
- Very strong customer satisfaction ratings in relation to the product.

A more recent 12 month case study (Oct 2008-Oct 2009) of an organization engaged in a completely different business domain (media) reported similar improvements in productivity and defect management (Middleton and Joyce, 2012). They proposed that the application of lean principles and practices are context dependent and not subject to a ‘cookbook’ type approach across companies and industries. Lean practices offer further advantages to organizations as the use of statistical process control may help in qualifying for CMMI level 4 certification. Although both studies reported various improvements as a result of the application of lean principles, the later study addressed work practices in greater detail and provided visibility into the power of statistical process control by graphically presenting work performance improvements. A notable observation presented in this more detailed study is the difficulty encountered in the application of the ‘Takt time’ concept to software development. The team did not measure against a required demand rate but instead observed increased throughput arising from cycle-time reductions. It is assumed that these cycle-time reductions resulted from various process improvements. The concept of a consistent software development unit of work was seen to be too difficult to establish. Therefore, the use of takt time to pace and control the production process was roughly simulated by using minimum market features. This resulted in concise cohesive software development iterations. These iterations, supported by daily meetings, created small feedback loops that provided momentum and control of the process (R5). This interpretation is consistent with the concept of “self-reinforcing virtuous cycles” (Charette, 2003), “punctuated equilibrium” (Sutherland, 2008) and “fast powerful feedback loops” (Reinertsen, 2005).

A consistent theme espoused by Middleton in his research is the cultural impact of the application of lean principles to software development within manufacturing organizations or companies serving the manufacturing community. He contended that
this approach to work serves to embed the language and values of manufacturing within the software development teams. Although not explicitly stated, one has to assume that the implication of a software team’s adoption of these values is that it increases their ability to focus on the needs of their customers/market.

An influential contribution to the growing body of knowledge on LSD was made by (Poppendieck and Poppendieck, 2003) in their proposal and analysis of a set of lean principles to support LSD. The emphasis on this work was to provide an application framework of twenty two practices that could enable practitioners to realize lean values. The principles proposed were: waste elimination (P1); learning amplification (P2); deferment of decisions (P3); optimal speed of delivery (P4); team empowerment (P5); integrity construction (P6) and systems thinking (P7). Poppendieck and Poppendieck (2007) expanded the nature of their earlier tactical software-centric work to incorporate the impact of the broader lifecycle required to incorporate strategic considerations espoused by lean thinking. Table 2-1 presents a codified list of the lean principles put forward by the various authors reviewed in this section.
<table>
<thead>
<tr>
<th><strong>Operations Management</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ohno (1988)</strong></td>
</tr>
<tr>
<td><strong>O1</strong>: Continual emphasis on waste reduction; <strong>O2</strong>: Flow; <strong>O3</strong>: Built-in quality; <strong>O4</strong>: Respect for people; <strong>O5</strong>: Caution in slow-growth environment; <strong>O6</strong>: Production levelling; <strong>O7</strong>: Autonomation; <strong>O8</strong>: Respond to needs; <strong>O9</strong>: Cost reduction</td>
</tr>
<tr>
<td><strong>Spear and Bowen (1999)</strong></td>
</tr>
<tr>
<td><strong>SB1</strong>: Detailed specification of every task; <strong>SB2</strong>: Unambiguous communication procedures between employees; <strong>SB3</strong>: Single pathway for every product; <strong>SB4</strong>: Strict adherence to scientific method for improvements</td>
</tr>
<tr>
<td><strong>Womack &amp; Jones (2003)</strong></td>
</tr>
<tr>
<td><strong>W1</strong>: Value; <strong>W2</strong>: Value Stream; <strong>W3</strong>: Flow; <strong>W4</strong>: Pull; <strong>W5</strong>: Perfection</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Product Development</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reinertsen and Shaheff (2005)</strong></td>
</tr>
<tr>
<td><strong>R1</strong>: Batch size reduction; <strong>R2</strong>: Tolerate necessary variability; <strong>R3</strong>: Maintain flow; <strong>R4</strong>: Pull; <strong>R5</strong>: Fast powerful feedback loops; <strong>R6</strong>: Avoid over-reliance on requirements; <strong>R7</strong>: Invest in flexibility; <strong>R8</strong>: Achieve adequate failure rates; <strong>R9</strong>: Understand economics of waste; <strong>R10</strong>: Avoid sub-optimization</td>
</tr>
<tr>
<td><strong>Liker and Morgan (2006)</strong></td>
</tr>
<tr>
<td><strong>L1</strong>: Establish customer-defined value to separate value added from waste; <strong>L2</strong>: Maximize design space(explore alternatives); <strong>L3</strong>: Levelled process flow; <strong>L4</strong>: Rigorous standardization; <strong>L5</strong>: Chief Engineer system; <strong>L6</strong>: Balance functional and cross-functional expertise; <strong>L7</strong>: Develop engineering competence; <strong>L8</strong>: Integrate suppliers; <strong>L9</strong>: Continuous improvement; <strong>L10</strong>: Build culture of relentless improvement; <strong>L11</strong>: Adapt technology to fit people and process; <strong>L12</strong>: Simple visual communication to align organization; <strong>L13</strong>: Support standardization and learning with powerful tools.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Project Management</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Charette (2003)</strong></td>
</tr>
<tr>
<td><strong>C1</strong>: Customer satisfaction; <strong>C2</strong>: Active customer participation; <strong>C3</strong>: Team effort; <strong>C4</strong>: Everything can change; <strong>C5</strong>: Domain solutions; <strong>C6</strong>: Avoid duplication; <strong>C7</strong>: 80% solution now preferable; <strong>C8</strong>: Minimalism; <strong>C9</strong>: Needs determine technology; <strong>C10</strong>: Measure product growth in features; <strong>C11</strong>: Use lean appropriately; <strong>C12</strong>: Worker development</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Lean Software Development</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hou (1995)</strong></td>
</tr>
<tr>
<td><strong>H1</strong>: Defined; <strong>H2</strong>: Configuration Management, <strong>H3</strong>: User Involvement, <strong>H4</strong>: Transparent, <strong>H5</strong>: Tailorable, <strong>H6</strong>: Rapid Development, <strong>H7</strong>: Scalable, <strong>H8</strong>: Defect-free, <strong>H9</strong>: Continuous improvement, <strong>H10</strong>: Manufacturability, <strong>H11</strong>: Supportability, <strong>H12</strong>: Scalable architecture, <strong>H13</strong>: Hardware/software codesign</td>
</tr>
<tr>
<td><strong>Middleton (2001a)</strong></td>
</tr>
<tr>
<td><strong>M1</strong>: Continual improvement; <strong>M2</strong>: Empowered workers; <strong>M3</strong>: Defect prevention; <strong>M4</strong>: Visual quality measures; <strong>M5</strong>: Automatic quality measurement devices(often self-developed)</td>
</tr>
<tr>
<td><strong>Poppendieck &amp; Poppendieck (2003)</strong></td>
</tr>
<tr>
<td><strong>P1</strong>: waste elimination; <strong>P2</strong>: learning amplification; <strong>P3</strong>: deferment of decisions; <strong>P4</strong>: optimal speed of delivery; <strong>P5</strong>: team empowerment; <strong>P6</strong>: integrity construction; <strong>P7</strong>: systems thinking</td>
</tr>
<tr>
<td><strong>Middleton, Flaxel &amp; Cookson (2005)</strong></td>
</tr>
<tr>
<td><strong>MF1</strong>: Small iterations; <strong>MF2</strong>: Requirements elicitation; <strong>MF3</strong>: Requirements chunking to support resource allocation; <strong>MF4</strong>: Units of work to enable Takt time establishment; <strong>MF5</strong>: Low distance between collaborators; <strong>MF6</strong>: Consistent roles, work practices; <strong>MF7</strong>: Defect prevention; <strong>MF8</strong>: Feedback on productivity &amp; errors; <strong>MF9</strong>: Impartial data collection; <strong>MF10</strong>: Multi-disciplinary teams;</td>
</tr>
<tr>
<td><strong>Middleton &amp; Joyce (2012)</strong></td>
</tr>
<tr>
<td><strong>MJ1</strong>: Levels of WIP; <strong>MJ2</strong>: Pull work only when needed; <strong>MJ3</strong>: Level out workload; <strong>MJ4</strong>: Stop to fix problems; <strong>MJ5</strong>: Continuous improvement; <strong>MJ6</strong>: Make process visible; <strong>MJ7</strong>: Ensure technology serves people &amp; process;</td>
</tr>
</tbody>
</table>

**Table 2-1: Lean principles from various sectors/authors**
## 2.3 Terminology

<table>
<thead>
<tr>
<th>LEAN TERM</th>
<th>DESCRIPTION</th>
<th>LSD EQUIVALENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Muda</td>
<td>Any activity that consumes resources but does not create value (Womack &amp; Jones, 2003)</td>
<td>Waste</td>
<td>Defects; Unnecessary features; Misunderstandings during handovers; Unavailable resources to address complex specifications; Idle resources due to work-in-progress delays;</td>
</tr>
<tr>
<td>Mura</td>
<td>Inconsistency (variability)</td>
<td>Process improvement</td>
<td>Reinertsen (2005) and Middleton and Sutton (2005) both allude to the reduction in variability of software development by reducing mistakes (waste) through continual process improvement. Many authors promote the use of smaller work batches (iterations) in order to reduce schedule variability.</td>
</tr>
<tr>
<td>Muri</td>
<td>Unreasonableness</td>
<td>Overwork</td>
<td>Increased workloads can result in defects being introduced to a system (Larman and Vodde, 2009)</td>
</tr>
<tr>
<td>Value</td>
<td>Capability provided to the customer at the right time at an appropriate price - as defined by the customer (Womack and Jones, 2003)</td>
<td>Customer-defined satisfaction with product features, timing and cost. May include many quality attributes including reliability, security, testability, reusability (McCall et al., 1977).</td>
<td>Middleton (2005) discussed extensive customer engagement to formulate requirements. He then describes use of the Kano technique to determine if customer’s satisfaction with the product was basic or expected (conformance to requirements) or exciting (fitness for purpose). These tacit unexpressed customer requirements constitute big ‘Q’ quality and are a major part of the value to the customer (Kano et al., 1994).</td>
</tr>
<tr>
<td>Value stream</td>
<td>Map every step from initial transformation of raw materials to delivery of final product in order to identify and tackle value and waste. Process efficiency is (value added time/lead time) * 100% so efficiency can be increased by waste reduction leading to lead time reduction. (Womack and Jones, 1993)</td>
<td>Implementation of software method</td>
<td>Each processing step, including quality assurance review, design meeting, code testing etc. constitutes the value stream and can be mapped to identify the value added, non value-added and necessary non-value added activities being performed.</td>
</tr>
<tr>
<td>Flow</td>
<td>Establish a process flow of value adding activities to delivery of product (value)</td>
<td>Software development process</td>
<td>Note that the original meaning of flow was to reduce batch sizes as small as possible to have a product flow through production out to customer. The analogy for software is not a product - it is a feature. Middleton (2005, 2012) acknowledged the difficulty of establishing a ‘standard feature’ but presented the use of feature sets or ‘minimum marketable features’ (MMF) to deliver value.</td>
</tr>
<tr>
<td>LEAN TERM</td>
<td>DESCRIPTION</td>
<td>LSD EQUIVALENT</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Pull</td>
<td>Setting up production to react to customer demand - begin with final customer and synchronize with product completion. Then work back through earlier processes (Ohno, 1988). No upstream work is delivered until immediate succeeding downstream operation signals a need (via Kanban)</td>
<td>Method (or software development process)</td>
<td>Example could be to use early adopter customer to act as a proxy for the market. Customer is enabled to request features and receive regular ‘drops’ or builds of product functionality. Similar to Scrum product backlog concept. Within the development process of a feature, process steps that have not been performed await signals of need from succeeding process steps (Hibbs et. al, 2011 p.113).</td>
</tr>
<tr>
<td>Perfection</td>
<td>Continuous improvement of standard work</td>
<td>Software process improvement</td>
<td>An example is a journey through the different capability levels of the ISO-15504 model in order to achieve level 5 which indicates an organization has processes defined for engineering, customers-suppliers, management, organization and support. The organization is adept at measuring these processes and introducing improvements in a disciplined fashion (Zahran, 1998)</td>
</tr>
<tr>
<td>Kaikaku</td>
<td>Radical introduction of an improvement to remove waste - also known as a breakthrough Kaizen (Womack &amp; Jones, 2003)</td>
<td>Major process change - such as reorganization or new method introduction.</td>
<td>Example might be a reorganization of software development teams from a matrix configuration to a projectized configuration (PMI, 2008)</td>
</tr>
<tr>
<td>Kaizen</td>
<td>Incremental improvement of an activity to create more value and reduce waste (Womack &amp; Jones, 2003)</td>
<td>Process improvement</td>
<td>Example may be the introduction of code reviews to improve standards compliance, thus increasing software maintainability and reducing defects.</td>
</tr>
<tr>
<td>Jidoka (Autonomation)</td>
<td>Enabling machines to detect defects and automatically stop working while alerting an operator to resolve the issue.</td>
<td>Build verification testing</td>
<td>Automated regression testing to verify system and suspend continuous integration of latest build if not 100% defect free.</td>
</tr>
<tr>
<td>JIT</td>
<td>Method to enable flow. A part is produced at the time it is needed. Requires upstream activities that have rapid set-up systems in order to respond to demand and downstream activities to have levelled their schedules to enable predictable flow. Ohno (1988) notes that the arrival of just enough inventory just in time enables waste reduction, promotes consistency and reduces the chances of employees being overburdened by workloads.</td>
<td>Lean software development</td>
<td>Iterative development of customer driven functionality (agile methods like Scrum or XP) which is linked to other aspects of the organization such as marketing, training, product management, sales, services and support.</td>
</tr>
<tr>
<td>LEAN TERM</td>
<td>DESCRIPTION</td>
<td>LSD EQUIVALENT</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SMED</td>
<td>Single-minute exchange of dies. Setting up a production line.</td>
<td>IDE version build</td>
<td>Creation of an integrated development environment for development activities (code/QA)</td>
</tr>
<tr>
<td>Visual control</td>
<td>Actual progress of work in relation to daily production plans is clearly visible.</td>
<td>Simple iteration goals</td>
<td>Maximize comprehension of immediate goals by delivering minimum marketable features in small iterations supported by daily meetings to enhance visibility and collaboration. Middleton (2011) described the use of Kanban boards to enhance process visibility and progress control.</td>
</tr>
<tr>
<td>Gemba</td>
<td>Place of work. Notion that continuous improvement should be initiated by the workers as they are best positioned to understand what impacts their job performance.</td>
<td>Regular retrospective</td>
<td>Regular meeting (usually at completion of iteration) for team to reflect on how to improve their processes. At another level, respect for the ‘Gemba’ is evident when teams are organized to take feedback from roles using work produced by other specialists. E.g. business analysis specification informed by architect/designer insights; design specifications informed by programmers/testers/technical writers.</td>
</tr>
<tr>
<td>Andon lights</td>
<td>Visual indicators displaying the status of production. Green means ok, yellow means operator wants help and red indicates line is stopped to fix a problem.</td>
<td>Information Radiators</td>
<td>(Middleton and Joyce, 2012) Another example is a flashing red light triggered by defect occurrences in build verification tests associated with continuous integration (Hibbs, 2008)</td>
</tr>
<tr>
<td>Kanban</td>
<td>Signalling system to facilitate ‘PULL’. An example is a card attached to a parts container indicating an upstream need for a part.</td>
<td>Kanban</td>
<td>Using sticky notes on a whiteboard to represent a work item (or process step, e.g. Design). A downstream process step can ‘pull’ upstream workitem outputs as needed.</td>
</tr>
<tr>
<td>Takt time</td>
<td>The rate at which items should be produced in order to meet demand</td>
<td>Cadence or rhythm of development</td>
<td>Achieved by timeboxing (Larman and Vodde, 2009)</td>
</tr>
<tr>
<td>Baka-yoke (Poka-yoke)</td>
<td>Mistake-proof. Device or procedure to prevent defect occurrence during order-taking or manufacture.</td>
<td>Defect prevention.</td>
<td>Many different mechanisms exist. One example is automated qualification testing against pre-defined coding standards that rejects non-standard code submitted to the configuration management system.</td>
</tr>
<tr>
<td>Baton Passing Zone</td>
<td>Ability for others in the production line to assist in handover of parts, when needed.</td>
<td>Inter-disciplinary collaboration</td>
<td>Core value of the Agile manifesto (Fowler &amp; Highsmith, 2001)</td>
</tr>
<tr>
<td>Five Why’s</td>
<td>Root-cause analysis. Often used with Ishikawa diagrams to facilitate asking ‘why?’ to the answer of a previous enquiry in order to determine the root cause for an issue.</td>
<td>Root cause analysis</td>
<td>Technique used in various methods. Cleanroom engineering is an example of a method that uses this technique to improve process (Hou, 1988)</td>
</tr>
<tr>
<td>Cells</td>
<td>Concept of putting all machines in a production line close together to promote one-piece flow</td>
<td>Development team configured in collaborative system. Names include ‘bullpen’ (US baseball term).</td>
<td>Grouping all members of a multi-disciplinary team in a ‘team space’ often with desks facing one another in u-shape.</td>
</tr>
<tr>
<td>LEAN TERM</td>
<td>DESCRIPTION</td>
<td>LSD EQUIVALENT</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Heijunka (Prod’n Levelling)</td>
<td>Downstream work must be levelled in order to reduce variations that would impact upon upstream work and impact upon JIT</td>
<td>Customer governance</td>
<td>Middleton (2011) explains that close collaboration with users was critical. Maintenance of smooth demand in upstream activities levels the work, thus enabling flow.</td>
</tr>
<tr>
<td>Standard Work</td>
<td>Description of a work activity. Contains the cycle time, takt time, sequence of steps and parts required.</td>
<td>Agreed development method (Larman, 2009)</td>
<td>A method-in-action (Fitzgerald, 2002). Carmel (1999) further expanded by stating that although a clearly-defined method was a centrifugal force in the establishment of distributed teams, it was recommended that local teams adapt the central method to their situation.</td>
</tr>
<tr>
<td>Cycle time</td>
<td>The rate at which items should be produced in order to meet demand</td>
<td>Recommended that cycle-time is reduced in order to enhance learning and implement improvements</td>
<td>Middleton &amp; Sutton (2005) explains that cycle time is time unit spends as WIP (from work start to release) and given the nature of software, a standard unit is difficult to establish. However, it still recommends constant attention to reduced cycle times to gain the benefits of same.</td>
</tr>
<tr>
<td>Stopping the line</td>
<td>Stopping production of an item. All downstream steps are suspended until the line is re-started at the point it was stopped.</td>
<td>Formal design reviews. More specifically in LSD, root cause analysis.</td>
<td>Formal design reviews. More specifically in LSD, root cause analysis.</td>
</tr>
</tbody>
</table>

Table 2-2: Lean terminology applied to LSD
2.4 Principles and values of LSD

This section presents a set of higher-order values that underpin lean principles and practices. In his work on learning organizations, Senge (1990) described practices as the activities that practitioners perform and principles as the ‘guiding ideas and insights’ that inform them on the rationale behind the activities that they select and implement. This is equivalent to the approach adopted in the method rationale analysis framework\(^2\), which proposes that a method fragment is performed by a practitioner in order to achieve a certain goal (Ågerfalk, 2006).

Table 2-1 presented lean principles proposed by different activists in LM, product development, project management and software development. The principle sets presented in this table were used to propose an underlying value being addressed by each principle. Using a hermeneutic interpretive process, these lean principles were synthesized into a set of values intended to abstract the beliefs and intentions of the principles.

A bottom-up approach was adopted to identify the candidate set of values. Principles were analysed in an iterative fashion in order to support the induction of values. To support the induction process, the literature used to source the principles was analysed further in order to deepen understanding of the meaning of the principle in the context of the paper in which it was presented. A description of each principle was established and these descriptions were used to produce definitions for an initial set of values. Each iteration of the analysis process resulted in related values being synthesized into higher-order values. Overlaps were encountered as aspects of different values are addressed in many of the principles. It was decided to consider the value that appeared to be most prevalent in each principle and to assign the principle to that value. This is considered reasonable as the motive was not to build an exact mapping of principles to values, but rather to use the principles as a basis to help uncover a representative set of lean values.

The iterative synthesis of values involved both convergence from narrow to broad (as similar values were grouped to a higher-order value) and in some cases divergence from high-level to specific. An example of this approach was the emergence and establishment of the value

\(^2\) The concepts that drive method rationale (values, goals and practices) are described in chapter 5.
‘Effective project management’ as underpinning a number of principles. A subsequent iteration resulted in this value being subsumed into a broader value: ‘Effective process’. The synthesis process was not always one of convergence. A final run-through of the values led to the identification of the value ‘Flow of value’. This had been part of the value ‘Effective process’ but it became clear that many researchers had singled out ‘flow’ as a significant aspect of lean thinking. Therefore, this concept was deemed of sufficient relevance to merit consideration as a specific lean value. An expanded version of the value analysis process is presented in appendix A. The proposed set of LSD values is presented in table 2-3. Linkage between values and their associated principles is shown through the principle codes.
<table>
<thead>
<tr>
<th>ID</th>
<th>Value</th>
<th>Description</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Customer value</td>
<td>Ensure customer defines value. Seek out user needs, not just requirements. Early delivery of value. Enable future needs (maintainability). Enable efficient deployment of product features. Provide value for money.</td>
<td>H3,H6,H10,H11,H12,H13,H14,MF1,MF2,MF10,L1,W1,W4,09,C1,C2,C3,C8,P4</td>
</tr>
<tr>
<td>V2</td>
<td>Remove waste</td>
<td>Promote systematic defect prevention. Eliminate rework through emphasis on customer needs. Holistic focus on quality that motivates the refactoring of legacy code to improve general product. Identify silos of waste present in value stream and improve processes. Promote reusability where appropriate. Take cognisance of the fact that waste can have different forms in different projects/work situations.</td>
<td>H8,M3,M7,M14,W2,0,1,03,C7,P1,R9</td>
</tr>
<tr>
<td>V3</td>
<td>Flow of value</td>
<td>Stabilize development process to enable levelled flow of value - manage necessary variability. Smooth demand from users. Pull value from user demands through lifecycle. Apply minimalism (scope items, teams, documents). Allocate resources only when needed.</td>
<td>M12,M13,L3,W3,02,06,C9,R1,R2,R3,R4</td>
</tr>
<tr>
<td>V4</td>
<td>Person focus</td>
<td>Empower individuals and teams. Promote continual worker development. Invest in multi-skilling and promote attitude to participate in multiple roles.</td>
<td>M2,O4,C13,P5,R7</td>
</tr>
<tr>
<td>V5</td>
<td>Continuous Improvement</td>
<td>Visible feedback on productivity. Reviews to propagate learning to wider organizations. Apply rigorous standardization to establish agreed baselines for improvement. Actively seek and manage obstacles through agreed processes, such as root cause analysis. Promote organizational learning and a culture of relentless improvement. Use simple powerful tools.</td>
<td>H9,MF1,M8,M11,4,19,L10,L11,L5,07,P7,SB4</td>
</tr>
<tr>
<td>V6</td>
<td>Product excellence</td>
<td>Promote strong design culture to avoid premature convergence on incorrect solution. Prototype different options to derive best approach. Deliver functionality in change-tolerant form. Build quality in through practices of jidoka and poka-yoke. Promote deep specialized knowledge of product and process. Promote a culture of excellence.</td>
<td>L2,L7,C11,P6,R8</td>
</tr>
<tr>
<td>V7</td>
<td>Embrace change</td>
<td>Defer commitment to scope. Encapsulate features and consider all options carefully. Facilitate emerging requirements.</td>
<td>C5,P3,R6</td>
</tr>
<tr>
<td>V8</td>
<td>Business environment</td>
<td>Be aware of why you are doing work. Ensure it adds value to your situation. Future-proof your approach to work against changing market conditions. Consider domain solutions rather than restricting product to one market sector. Don’t force lean approach where inappropriate to business needs. Don’t optimize locally to the detriment of other aspects of the business.</td>
<td>H5,05,08,C6,C12,R10</td>
</tr>
<tr>
<td>V9</td>
<td>Data driven decisions</td>
<td>Impartial data collection to drive decision making and reduce cost of meetings and disagreements. Rigorous scientific approach to continuous improvement.</td>
<td>MF9</td>
</tr>
<tr>
<td>V12</td>
<td>Effective use of technology</td>
<td>Technology must serve a particular need. Configuration management systems and quality measurement devices are examples of key facilities. Pursue automation by using technology to enable workers to effectively adjust system when necessary.</td>
<td>H2,M5,M17,L11,07,C10</td>
</tr>
</tbody>
</table>

Table 2-3: LSD Values and associated principles
2.5 Summary

A set of LSD values were proposed as a result of the work reported in this chapter. These values were induced from principles identified by various contributors to lean thinking (LT) across a number of domains. The identification of a representative set of commentators on LT was approached from a genealogical perspective. Academic module materials UL (2005), lean training\(^3\) and Ohno (1988) formed the core searches in operations management. The next phase was to generalize lean operations. Womack and Jones (2003) served as a launching pad to conduct citation searches for relevant papers, websites, books and conference proceedings. These searches were influenced by interest in the application of LT to software development. The proposal that Scrum and Lean were two separate approaches to address complex adaptive systems (Sutherland, 2008) prompted investigation into product development. Inconsistencies in reporting Charette’s lean development method highlighted confusion about the relationship between lean and agile in software development and motivated a review of his work in project management and lean thinking. Finally, the software focus directed the search to review sources related to the term ‘lean software development’.

A hermeneutic interpretive process was followed to induce a set of LSD values from the various lean principles uncovered in the literature. In order to support the reader’s understanding of this subject in the context of software development, section 2.2 presented a table mapping lean terminology to its application in an ISD context.

---

\(^3\) The application of lean principles to software development. This was a 2 day lean concepts course delivered in NUIG, Ireland and tailored for members of the ALPS (Agile and Lean Project Management and Scaling) project. ALPS is part of the LERO (www.lero.ie) CSET research program funded by Science Foundation Ireland.
Chapter 3- Scrum Practices

3.1 Introduction

Figure 3-1 illustrates the detailed work packages that form the second major body of work required to construct the conceptual model.

Chapter 2 established the first of the three legs that form the basis for the candidate LDScrum model: LSD values. This chapter proposes a set of core Scrum practices which forms the second leg of the model. A partial list of the goals that form part of the final leg of the model is also identified and each of the core Scrum practices is associated with the achievement of particular goals.

Section 3.2 introduces the Scrum framework. An examination of the characteristics of successful new product development (NPD) teams and ASD principles is used to propose a set of goals to
be achieved through the application of this agile method. Scrum is decomposed into a set of twelve core practices that reflect guidelines on how the method should be performed.

Section 3.3 presents a more detailed examination of Scrum practices and outlines forty eight supplementary Scrum practices that have been identified from literature. Associations to the identified goals are proposed for each of these practices. Practices are presented in five categories: general considerations; pre-sprint practices; sprint practices; post-sprint practices and release management.

Finally, the chapter concludes by synthesizing the analysis of supplementary practices up to their ‘parent’ core practice. This results in a set of core Scrum practices that are associated with the performance of Scrum in ‘regular’ (or collocated small-team) environments. This table of associations forms a baseline that may be extended to address Scrum in other situations. Chapter 4 extends this set to the application of Scrum in GSD.
3.2 Origin and Overview of Scrum

“...”

(Schwaber, 1995, p. 10)

This section describes NPD characteristics and ASD principles proposed in the Agile Manifesto. It then provides an overview of the Scrum framework and proposes twelve core practices that are used to perform Scrum. Finally, it concludes by providing a high-level overview relating the twelve core Scrum practices to NPD characteristics.

3.2.1 The origins of Scrum - New Product Development (NPD) characteristics

The use of the tactics of rugby union by Takeuchi and Nonaka (1986) as an analogy to describe an innovative form of new product development inspired Jeff Sutherland and Ken Schwaber to create the Scrum development method. This method was devised to address their concerns about the unpredictable nature of ISD. Schwaber (1995) noted that many ISD methods are applied as theoretical (or deductive) constructs that rely on well understood internal processes derived from...
first principles, whereas ISD methods are in fact empirical (or activities that rely upon observable experiences). He stated that the inner workings of ISD methods require measurement and controls as their application is unpredictable. Scrum emphasises a management framework and does not prescribe technical practices (Schwaber, 1995). The distinction between methods conceived from a theoretical viewpoint as against the empirical approach adopted by Scrum is most evident when methods are applied to very complex projects. The lack of predictability and repeatability associated with ISD is aggravated in complex projects by the ‘noise’ of constant change. The empirical foundation of Scrum addresses such situations more effectively than traditional ISD methods (Highsmith, 2002).

The use of a rugby Scrum analogy to describe high-performing new product development (NPD) teams inspired the development of the Scrum framework. This work proposed that the combination of six management characteristics results in very positive and effective product development: built-in instability, self-organizing project teams, overlapping development phases, multi-level learning, subtle control and organizational transfer of learning (Takeuchi and Nonaka, 1986). Setting challenging targets and releasing control of how these targets should be met introduces a tension and instability that empowers teams, promoting creativity and innovation. Self-organization incorporates autonomy, self-transcendence and cross-fertilization. Autonomy empowers the team to manage their situation without external interference. Self-transcendence is evident when teams practice continuous improvement through establishment and extension of goals in pursuit of the higher-order challenges set down by senior management. Cross-fertilization occurs as teams share knowledge. Earlier agile settings promoted co-located teams. It is likely that a motivation for such a configuration was the dissemination of tacit knowledge via socialization (Turban et al., 2007). Cross-fertilization is both leveraged and augmented by overlapping development phases that promote collective responsibility and the reduction of potential delays. Such delays may be due to bottlenecks and misunderstandings associated with the ‘over the wall’ hand-off approach associated with division of labour using separate specialized development phases. Although the decomposition of work into phases is cited as a reason for method usage due to the reduction in development complexity and facilitation of project management through the division of labour (Fitzgerald et al., 2002), it is
also considered a source of delays and waste, especially in distributed development (Grinter et al., 1999). Multi-level learning encompasses the notion of learning being promoted at multiple levels (individual, group, and corporation) and that individuals seek learning across functions (such as ISD and marketing). Subtle control refers to the management of self-organizing teams by permitting freedom within boundaries that are marshalled through non-invasive management techniques that are tolerant and reflective of the uncertain non-linear nature of new product development. Organizational dissemination of learning is enabled through standardization and sharing via formal artefacts (e.g. project post-mortem (PMI, 2008)) and judicious assignment of experienced individuals to appropriate projects (Takeuchi and Nonaka, 1986).

<table>
<thead>
<tr>
<th>Code</th>
<th>Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPD-1</td>
<td>Built-in instability</td>
</tr>
<tr>
<td>NPD-2</td>
<td>Self-organizing project teams</td>
</tr>
<tr>
<td>NPD-3</td>
<td>Overlapping development phases</td>
</tr>
<tr>
<td>NPD-4</td>
<td>Multi-level learning</td>
</tr>
<tr>
<td>NPD-5</td>
<td>Subtle control</td>
</tr>
<tr>
<td>NPD-6</td>
<td>Organizational transfer of learning</td>
</tr>
</tbody>
</table>

Table 3-1: New Product Development principles that inspired Scrum founders (Takeuchi and Nonaka, 1986, Schwaber, 1995)

In his introductory publication to the Scrum method, (Schwaber, 1995) invoked complex adaptive systems theory (CAS) stating that Scrum enables a team to maintain order within the development process while embracing change within the environment. Wang and Conboy (2009) specified this relationship further by comparing particular CAS concepts with aspects of agility. They stated that the notion of inter-connected autonomous agents that can ‘flock together’ to react and resolve an external force is analogous to an ASD team that is empowered to direct its activities and consists of individual members that share their experiences and skills. The team is then in a position to come together in order to resolve unforeseen issues (i.e. adapt to change). This is consistent with the aforementioned characteristics of self-organization and overlapping development phases. A self-directed ASD team consisting of autonomous individual members who share each other’s experiences and skills reflects autonomy and cross-fertilization. Grouping together to resolve development issues indicates an ability to overlap development phases as analysis/design/code and test may be applied to overcome an issue. The CAS concept of ‘edge of
chaos’ is described as that state where a team (or organization) can experiment and adapt to situations freely but retain a limited structure in order to avoid falling into disarray. This equates to the aforementioned characteristics ‘built-in instability’ and ‘subtle control’. Finally, ‘emergence’ relates to continuous improvement or new learning leading to new behaviour that indicates that a team has adapted and evolved. This concept relates to ‘multi-level learning’ and when extended beyond the team may also apply to ‘organizational transfer of learning’. Given that NPD characteristics formed part of the inspiration for the Scrum method, it seems reasonable to propose that each of these six characteristics should be considered a goal to be pursued by Scrum teams. Section 3.2.4 provides additional details on how the effective application of Scrum may result in these characteristics being evident in an ISD team.

3.2.2 Agile software development (ASD) principles

Following publication of the ‘Agile manifesto’ (Fowler and Highsmith, 2001), many reviews of agile methods declared Scrum to be one of the primary ASD methods (Abrahamsson et al., 2003; Cohen et al., 2004; Highsmith, 2002; Dyba and Dingsoyr, 2008). The four core values proposed by the agile manifesto are supported by twelve guiding principles. ASD methods typically follow some of these principles (Conboy and Fitzgerald, 2004). Table 3-2 presents an identifier and description of each of these principles. It seems reasonable to leverage these principles as potential goals to be achieved by the application of ASD. As such, the position is taken to augment the aforementioned six NPD characteristics with these twelve principles in order to form a set of eighteen goals that may be achieved by the application of Scrum.
<table>
<thead>
<tr>
<th>Code</th>
<th>Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</td>
</tr>
<tr>
<td>A-2</td>
<td>Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.</td>
</tr>
<tr>
<td>A-3</td>
<td>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
</tr>
<tr>
<td>A-4</td>
<td>Business people and developers must work together daily throughout the project.</td>
</tr>
<tr>
<td>A-5</td>
<td>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</td>
</tr>
<tr>
<td>A-6</td>
<td>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</td>
</tr>
<tr>
<td>A-7</td>
<td>Working software is the primary measure of progress.</td>
</tr>
<tr>
<td>A-8</td>
<td>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</td>
</tr>
<tr>
<td>A-9</td>
<td>Continuous attention to technical excellence and good design enhances agility.</td>
</tr>
<tr>
<td>A-10</td>
<td>Simplicity – the art of maximizing the amount of work not done – is essential.</td>
</tr>
<tr>
<td>A-11</td>
<td>The best architectures, requirements, and designs emerge from self-organizing teams.</td>
</tr>
<tr>
<td>A-12</td>
<td>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.</td>
</tr>
</tbody>
</table>

Table 3-2: Agile principles (Fowler, 2001)

### 3.2.3 The Scrum Framework - core practices

Scrum is a framework of simple rules that enable continuous inspection and adaptation. The lifecycle of a Scrum project is comprised of a series of development ‘sprints’ or iterations that are bounded by an initial planning phase and a final closure phase (figure 3-2). Planning enables both architectural and scope concerns to be addressed and the closure phase incorporates release management. The initial and final phases are predictable and theoretical. The empirical nature of Scrum is evident in the ongoing iterations or ‘sprints’ that adapt to feedback and change throughout the project (figure 3-3). This nature relies upon transparency, inspection and adaptation. Transparency stipulates that the process is understood and that a standard language is used and accepted by team members and observers. Inspections of artefacts are performed to uncover defects or invalid variances and adaptations to the process are performed if deemed necessary. In core Scrum practices, inspection and adaptations may occur at the Sprint planning meeting, daily Scrum, sprint review meeting and the sprint retrospective (table 3-3). A notable
aspect of this approach is the inclusion of sales and marketing who have sometimes been viewed as obstacles to traditional development projects. The method embraces change by enabling the development team to both react and promote changes as the information system evolves (Sutherland and Schwaber, 2007; Sutherland and Schwaber, 2011).

Figure 3-3: Scrum Framework (Deemer et al., 2010)

Sutherland and Schwaber (2011) recommend Scrum as an effective container for other methods. Fitzgerald et al. (2006) describe the effective use of Scrum to plan and manage appropriate XP technical practices. Although they question the premise that the implementation of agile methods should not be decomposed into piecemeal selection of practices, they report the effective tailoring of Scrum and XP differently. Scrum tailoring involves the use of additional or supplementary practices to enrich the core set of practices. XP tailoring results in the application of a subset of the core set of practices. The founders of Scrum state that although it is possible to pick and choose certain core practices or omit certain roles, the Scrum framework should be used in its entirety. In cases where roles or practices have been omitted, the resultant framework should not be called Scrum (Sutherland and Schwaber, 2011). The position of this thesis is that the use of supplementary practices to deepen consideration of each core Scrum role and practice
upholds the guideline of using the complete framework. A set of ‘core’ Scrum practices are presented in Table 3-3 reflecting published reviews of Scrum from its first introduction (Schwaber, 1995), through a major ASD contribution (Highsmith, 2002) and up to the most recent definition of the framework by its founders (Sutherland and Schwaber, 2011)\(^4\). Each core Scrum practice is identified by an ID prefixed by ‘CS’. CS-7 was initially identified as ‘Daily Scrum’ but has been broadened by this author to reflect general Sprint management activities including Daily Scrum. An additional core practice, CS-8 ‘Sprint-technical activities’ was not proposed by any of the sources as technical practices are not explicitly recommended by Scrum. It was added by this author to reflect the fact that in order to do ISD using Scrum, technical activities must be performed. Certain technical practices support Scrum goals and it is deemed useful to include this category as a core Scrum practice in order to more fully explore the relationship between LSD and ASD.

\(^4\) This guide is updated regularly so this version reflects the most recent views of Sutherland and Schwaber at the time of writing
<table>
<thead>
<tr>
<th>ID</th>
<th>Practice</th>
<th>Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-1</td>
<td>Scrum Team Configuration</td>
<td>A Scrum team comprises a product owner, Scrum-master and developers. This team are cross-functional and self-organizing</td>
<td>Self-organizing team</td>
</tr>
<tr>
<td>CS-2</td>
<td>Initial planning (and ongoing) - create or update product backlog</td>
<td>Create product backlog - list of features needed in product.</td>
<td>Prioritize high-level features to be constructed</td>
</tr>
<tr>
<td>CS-3</td>
<td>Sprint planning - create release backlog</td>
<td>Determine list of features to be included in first release of product</td>
<td>Manage stakeholder expectations</td>
</tr>
<tr>
<td>CS-4</td>
<td>Sprint planning - create sprint backlog</td>
<td>Determine features to be developed in Sprint. Determine and estimate tasks to be performed.</td>
<td>Empower team and enable control of sprint</td>
</tr>
<tr>
<td>CS-5</td>
<td>Sprint planning - define sprint goal</td>
<td>High-level business purpose that provides rationale for Sprint features - much like a project charter (PMI, 2008)</td>
<td>Enable team to ‘perform’ by focussing on prime reason for their work rather than blindly following process (Tuckman, 1965)</td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - daily Scrum meeting and other general activities</td>
<td>Short meeting to identify and plan the removal of any impediments to team’s work. Enable information sharing and learning. Monitor progress.</td>
<td>Promote collaboration</td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td>Although technical practices are not suggested as part of the framework, certain technical practices support many Scrum goalscoli</td>
<td>Support collaboration, technical excellence, sustainability</td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td>X-axis = Days of sprint; Y-axis = work-hours remaining; Simple visual control of work remaining. Updated every day following re-calibration of remaining time after daily Scrum. Tracks downward to zero</td>
<td>Daily feedback; Inaccurate estimation visibility; Performance monitoring.</td>
</tr>
<tr>
<td>CS-10</td>
<td>Post-Sprint - review meeting</td>
<td>Demonstrate features constructed. Review technical work completed</td>
<td>Project management : product owner understands current completed. Updates product backlog to reflect learning.</td>
</tr>
<tr>
<td>CS-11</td>
<td>Post-Sprint - team retrospective</td>
<td>Inspect how team performed in Sprint. Similar to project ‘lessons learned’.</td>
<td>Continuous improvement</td>
</tr>
<tr>
<td>CS-12</td>
<td>Definition of Done</td>
<td>All the team agree on what constitutes a ‘done’ product increment. Item must be ‘done’ to be potentially shippable. Team may increase done qualification criteria as it matures.</td>
<td>Promote collaboration: Transparency to support inspection. Self-organization. Focus on quality.</td>
</tr>
</tbody>
</table>

Table 3-3: Core Scrum practices (Schwaber 1995; Highsmith 2002; Sutherland and Schwaber 2011)
3.2.4 The relationship between Scrum practices and NPD characteristics.

The core Scrum practices enable a project to exhibit the characteristics of innovative new product development. An established team configuration promotes subtle control and team self-organization. A product owner provides the ‘single point of accountability’ for product features, shielding the team from interference by multiple stakeholders. The Scrum-master promotes team self-organization by guiding without directing. The development team contains a heterogeneous set of skills. However the only role permitted in this team is ‘developer’ thus promoting the characteristic of overlapping development phases. Practices such as the creation of sprint backlog and daily Scrum promote and facilitate self-organization among the development team. Permitting the team to set the amount of work they may achieve in a sprint, combined with feedback mechanisms such as sprint backlog graph and sprint review facilitate a team’s understanding of their velocity and have the potential to encourage self-transcendence as they accelerate their velocity. Freezing requirements within a sprint exhibits subtle control. However, permitting unlimited change to the product backlog builds in controlled instability by allowing the team to work in an orderly manner within an extremely volatile environment. Sprint reviews promote transfer of information to other parts of the organization as this meeting informs stakeholders about the increment from a technical and business perspective. Team retrospective promotes multi-level learning as individuals reflect upon their performance and the team’s performance. Arguably, the Sprint retrospective may lead to process improvements and updated standards, thus enabling transfer of information to other parts of the organization. However, such behaviour is not recommended currently in retrospective activities.

3.3 Supplementary Scrum practices.

In order to probe further into the Scrum method, additional detail on the application of the above core practices is supplied by examining contributions from various sources. In many cases, these supplementary practices represent a variation on one of the core practices outlined above. It should be noted that any list of supplementary practices is unlikely to be exhaustive. For
example, Mike Cohn’s accompanying foreword to (Kniberg, 2007) cautioned readers to be conscious of the influence of context in the application of Scrum practices.

“There is not and never will be a list of ‘Scrum Best Practices’ because team and project context trump all other considerations. Instead of best practices, what we need to know are good practices and the contexts in which they were successful.”

(Cohn foreword in Kniberg, 2007, p. iii)

A synthesis of additional Scrum practices and their purpose in ISD projects is presented in tables 3-4 to 3-9. This synthesis is derived from an iterative analysis process of practice sets gleaned from papers and books that examine the Scrum framework. Each table addresses a particular category of practices, describing each practice and proposing associations between the application of the practice and the potential achievement of NPD and ASD goals. Categories address general considerations and practices associated with different phases of a Scrum iteration.

3.3.1 General Considerations

General practices that describe team setup and framework application are listed in table 3-4. A key tenet of Scrum is its transparency and this is supported by simple rules to promote consistent comprehension and application of the framework. Earlier versions of (Deemer et al., 2010) emphasised this simplicity by highlighting a ‘rule of three’ to explain that the framework involved a team that was limited to three roles, practised three key ceremonies and utilized three artefacts. Cohn (2010) emphasized the need for small cross-functional teams. He cited studies claiming evidence of ‘hiding’ or ‘social loafing’ when individuals feel that somebody else can cover for them. Cohesion is promoted in smaller teams due to increased interaction leading to the collective responsibility that is often evident through the ‘overlapping of phases’. Conversely, larger teams may result in greater specialization leading to silos of work that require increased coordination costs and reduced information-sharing.

Allocation of the entire development responsibility for a particular feature set to a team can prove effective. Such encapsulation of feature development offers potential to achieve the same
advantages as use of this concept can provide in object-oriented development: ‘high-cohesion’ and ‘low-coupling’. Clarity of purpose can lead to increased commitment to value creation. Reduced dependencies on component teams and their associated sprints reduce the risk of delays due to integration difficulties or planning challenges and bottlenecks. In order to nurture self-organization, selection and de-selection of team members is critical. Cross-fertilization demands consideration of diverse technical and domain skills. This is one aspect of the ‘subtle control’ characteristic that may be employed by management to promote effective self-organization. Within the Scrum environment, subtle control may be practised by the Scrum-master in how he/she engages with the dynamics within the development team. Cohn (2010) presented a practice that leverages a model of interconnected concepts in self-organization proposed by (Eoyang, 2001): containers, differences and transforming exchanges (CDE). This stipulates that if a Scrum-master is aware of issues within a team, he/she may alter the ‘container’ bounding a development team in order to influence how it self-organizes. Alteration of containers can range from a physical change such as the reduction of cubicle walls to a less tangible change such as the inclusion of the team in a community of practice. Differences within a team influence the manner in which it self-organizes so increasing or decreasing differences can impact this process e.g. asking probing questions in order to promote constructive dissent. Finally transforming exchanges refer to the manner in which a team self-organizes to address a challenge. For example, ongoing speeches related to concerns about product marketability by a product manager may motivate or de-motivate a team depending upon the situation. Altering the frequency of such an exchange may have a positive impact on the team’s ability to self-organize. Having team’s work together for a long time and dedicated to one project promotes relationship building as well as reducing waste caused by juggling multiple conflicting priorities. Learning occurs at the ‘team’ level as increased understanding and commitment to working together within a domain leads to increased performance (Tuckman, 1965). From a project management perspective, such a practice resembles a ‘projectized’ team configuration and is recommended for R&D projects (PMI, 2008). Although many general considerations presented here address the management aspects of the framework, a technical consideration is the automation of tests. Managing testing debt enables effective regression testing as increased features emerge from sprints. Automated tests support rapid and safe software builds by contributing to test suites that
may form integrated build verification tests to enable dynamic integration of features from earlier sprints. Consideration and attention to time-consuming activities supports sustainable work processes while attention to continuous verification of both the increment and its non-intrusive integration into the larger release adheres to the principle of working software being the primary measure of progress.

<table>
<thead>
<tr>
<th>Code</th>
<th>Scrum Practices</th>
<th>Description</th>
<th>Goal/Benefit</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-1</td>
<td>Rule of three (roles, ceremonies &amp; artefacts)</td>
<td>Product owner, Scrum master, team; sprint planning, review and daily Scrum, product backlog; sprint backlog and burn-down chart.</td>
<td>Ease of understanding of method</td>
<td>A-10</td>
</tr>
<tr>
<td>S-2</td>
<td>Small teams</td>
<td>Maximum of 10-12 people</td>
<td>Less coordination needed - enable informal communication; Less social loafing; less specialization (causing hand-offs and decreased learning); Increased morale;</td>
<td>NPD-3</td>
</tr>
<tr>
<td>S-3</td>
<td>Favour feature teams</td>
<td>Organize teams so that they are fully responsible for a feature</td>
<td>Reduced risk of poor component integration; Reduced handoffs; Increased focus on feature delivery;</td>
<td>A-1; A-5</td>
</tr>
<tr>
<td>S-4</td>
<td>Nurture self-organizing teams</td>
<td>Team determines how best to make decisions and allocate responsibilities. This is influenced by management considerations of 'containers', 'differences' and 'transforming exchanges'.</td>
<td>Empowerment of team</td>
<td>NPD-2; NPD-5; A-11</td>
</tr>
<tr>
<td>S-5</td>
<td>Single project assignment</td>
<td>Team members exclusively work on one project</td>
<td>Clear prioritization</td>
<td>A-1</td>
</tr>
<tr>
<td>S-6</td>
<td>Whole-team responsibility</td>
<td>Collective responsibility for success</td>
<td>Increased quality</td>
<td>NPD-2; A-9</td>
</tr>
<tr>
<td>S-7</td>
<td>One product owner</td>
<td>Single point of accountability</td>
<td>Rapid decision making (enabling change approval)</td>
<td>A-2</td>
</tr>
<tr>
<td>S-8</td>
<td>Role merging - reduce handoffs</td>
<td>Team all work together on backlog item with no handoff</td>
<td>Less specialization (causing hand-offs and decreased learning)</td>
<td>NPD-2; NPD-3; NPD-4</td>
</tr>
<tr>
<td>S-9</td>
<td>Keep team together for long periods</td>
<td>Enable team to effectively grow and relate well to one another</td>
<td>Continual learning</td>
<td>NPD-4</td>
</tr>
<tr>
<td>S-10</td>
<td>Involve widely</td>
<td>Team members included in all activities e.g. All developers in product backlog story writing</td>
<td>Encourage commitment</td>
<td>NPD-3; A-5; A-11</td>
</tr>
<tr>
<td>S-11</td>
<td>Share knowledge</td>
<td>Use techniques such as communities of practice to promote knowledge sharing</td>
<td>Continual learning</td>
<td>NPD-4; NPD-6</td>
</tr>
<tr>
<td>S-12</td>
<td>Manage testing debt</td>
<td>Keep improving test situation (automating) to make testing more manageable as increased features from Sprints could lead to unsustainable manual testing</td>
<td>Enable quality</td>
<td>NPD-2, A-7, A-8</td>
</tr>
</tbody>
</table>

Table 3-4: Supplementary Scrum Practices - General Considerations (Sutherland, 2006; Deemer et al., 2010; Cohn, 2010)
3.3.2 Sprint planning

Initial planning of a Scrum project involves establishing a shared understanding within the team about the vision, rationale and objectives of the project. The artefact that is used to facilitate this process is the product backlog. This is a flexible list of requirements ranked in order by the product owner. The dynamic nature of the list is reflected in its continual refinement throughout the project. The priority of items is updated as new requirements emerge. To support the ranking process, backlog features may be described as user stories. User stories from certain items may be sufficiently detailed to facilitate technical discussions with the development team during sprint planning. This follows the Agile Manifesto value of favouring discussion over the handover of detailed specifications. An additional level of granularity that may be added by the product manager to relevant user stories in order to support item clarification is ‘conditions of satisfaction’. These may form the basis for acceptance tests during the sprint. Other items will require progressive elaboration during the project before they will be available for development.

Cohn (2010) used the acronym ‘DEEP’ backlog to describe an effective product backlog i.e. Detailed appropriately so that enough top-ranked items are available for a sprint; Estimated so that it may enable planning; Emergent to embrace the dynamism of requirements changes; Prioritized by a customer value rating to enable the development team to deliver highest value with each sprint. Using high-level estimates to support planning until progressive elaboration and refinement of item user stories facilitates greater estimation accuracy is similar to the project management practice of ‘rolling wave planning’ (PMI, 2008). The ranking of product backlog items using customer value and development estimates may also take risk factors into account. Certain practices associated with product backlog management and sprint planning are listed in table 3-5 below.

The first sprint may begin once some product backlog items are available in appropriate detail. Before sprint commencement, the core practice of ‘sprint planning meeting’ is used by the Scrum team to pull items from the top of the product backlog and determine how they will be completed. Sutherland and Schwaber (2011) recommended that this be a two-part meeting to determine the ‘what’ and ‘how’ of the forthcoming sprint. Firstly, the development team select items to be developed from the product backlog. This is facilitated by the aforementioned DEEP
backlog. In the second part of the meeting, the development team conducts a high-level design in order to be confident that the work can be achieved within the sprint. This self-organization enables the team to maintain a sustainable workload. Early sprint work will be decomposed to a granularity of single-day tasks (although Woodward et al. (2010) suggested ‘story points’ as an alternative schedule breakdown technique). The development team complete sprint planning activities by establishing a sprint goal. This focuses the team on the rationale for delivering the product increment at the end of the sprint. Comprehension of the overarching goal to be achieved can help a team develop and sustain a mode of effective performance (Tuckman, 1965). From a planning and stakeholder management perspective, the sprint goal may be a milestone in a product roadmap (Sutherland and Schwaber, 2011).

<table>
<thead>
<tr>
<th>Code</th>
<th>Scrum Practices</th>
<th>Description</th>
<th>Goal/Benefit</th>
<th>Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-13</td>
<td>Backlog item estimation</td>
<td>To aid prioritization - Use Scrum team - Estimations of 10 developer days are ready for Sprint planning</td>
<td>Enable effective coordination. Meet customer’s needs. Assist in sprint planning.</td>
<td>A1; A-10</td>
</tr>
<tr>
<td>S-14</td>
<td>Progressive refinement of plans</td>
<td>Early plan captures essence of features - not specifics.</td>
<td>Minimizes investment; Enables deferred decision making to optimal time; Allows changes to plan</td>
<td>NPD-1; A-2</td>
</tr>
<tr>
<td>S-15</td>
<td>User stories</td>
<td>Simple template shifting focus from written documentation to discussion</td>
<td>Promotes communication on requirements</td>
<td>A-6</td>
</tr>
<tr>
<td>S-16</td>
<td>DEEP backlog</td>
<td>Detailed appropriate user stories, Estimated, Emergent, Prioritized</td>
<td>Enable planning</td>
<td>A-1; A-2</td>
</tr>
<tr>
<td>S-17</td>
<td>Unprotected product backlog</td>
<td>Product manager can change backlog not committed to Sprint in whatever way they wish. Always a Sprint away from getting their latest wishes</td>
<td>Provides acceptance of need to embrace change</td>
<td>NPD-1; NPD-5; A-2</td>
</tr>
<tr>
<td>S-18</td>
<td>Only pull stories into a sprint if they can be completed</td>
<td>Ensure sufficient planning on user story completed before it gets into a sprint</td>
<td>Avoid sprint extension</td>
<td>NPD-2; NPD-5; A-8</td>
</tr>
</tbody>
</table>

Table 3-5: Supplementary Scrum Practices - Initial Planning and Sprint planning (Sutherland, 2008; Deemer et al., 2010; Cohn, 2010)
3.3.3 **Sprint**

“The heart of Scrum is a Sprint, a time-box of one month or less during which a ‘Done’, useable, and potentially releasable product increment is created.”

(Sutherland and Schwaber, 2011, p. 8)

The core practices of daily Scrum and sprint backlog graph enable continual inspection and adaptation during development. Consistent sprint lengths enable the team to gain a better understanding of its capacity to deliver user stories. This enables it to better plan and to meet stakeholder expectations. Although it is good practice to consistently test features as a product is being built, it is wise to consider carefully how the backlog items will suit this approach. Mixing backlog items to ensure that certain items are completed early in the sprint can help avoid bottlenecks or handoff-related delays near the end of the sprint. The visual progress chart (or variations thereof) can assist in exposing increased work-in-progress due to poor item completion rates. A more advanced form of Scrum that incorporates lean principles and encourages flow extends this concept further. The introduction of a Kanban or pull approach combined with limits of how many items may be in a particular stage (such as ‘in-progress’) can force the team to resolve bottlenecks. This helps to reduce the risk of having many incomplete items at the end of the sprint (Kniberg and Skarin, 2010; Ikonen et al., 2011). Acceptance test driven development scopes the development of an item by linking the user story to the conditions of satisfaction specified by the product owner when considering the item as part of a releasable product increment. Although the product owner can change items on the product backlog at any time, it is also useful for the development team to be involved in refining (‘grooming’) the backlog to add detail such as estimates or user story clarifications.
Although Scrum is a framework used to manage other methods, certain technical practices are often associated with Scrum projects. Sutherland and Schwaber (2007) explained that the first use of Scrum in 1993 used all extreme programming (XP) engineering practices and that these were shared with Kent Beck as he commenced his formal definition of the XP method (Beck, 1999). The technical practices outlined in table 3-7 resemble particular XP practices. Developing a test prior to designing and coding a user story assists the development team in fully comprehending the work to be performed. The ASD principle of simplicity in order to frame activities to only address relevant work is supported by this approach. Pair programming promotes collective ownership and improves quality by advocating better designs through...
sharing of abstractions and linking analysis to design (Wasserman, 1996). Quality is also supported by this practice through continuous peer review and assistance in the identification and implementation of refactoring opportunities. Such an approach facilitates the overlapping of phases and favours face-to-face discussion. It is likely that both peer review and refactoring address continuous attention to technical excellence and design and should result in peer-learning. Product backlog items or associated user-stories should be allocated with clear interfaces as more detail emerges on these items. This low-coupling of ‘packets’ of work can ease design, testing and maintenance (Cohen et al., 2004). Such encapsulation can be seen as subtle control as teams are obligated to conduct design-by-interface but are empowered to evolve their own feature thus following the ASD guideline that the best architectures and designs emerge from self-organizing teams.

<table>
<thead>
<tr>
<th>Code</th>
<th>Practice</th>
<th>Description</th>
<th>Goal/Benefit</th>
<th>Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-30</td>
<td>Frequent build</td>
<td>Frequently integrate work into testable, documented ‘whole’. May be enabled</td>
<td>Early error detection (reduce technical debt). Enable progress visibility.</td>
<td>A-3; A-7</td>
</tr>
<tr>
<td>S-31</td>
<td>Low-coupling packets</td>
<td>Work allocated with clear interfaces</td>
<td>Ease of development (design, coding, testing, integration and maintenance). Positive design influence (Conway’s law).</td>
<td>NPD-5; A-5; A-11</td>
</tr>
<tr>
<td>S-32</td>
<td>Constant testing</td>
<td>Consistently test product as it is built.</td>
<td>Early error detection (reduce technical debt).</td>
<td>A-9</td>
</tr>
<tr>
<td>S-33</td>
<td>Constant documentation</td>
<td>Consistently document product as it is built.</td>
<td>Facilitate testing and maintenance. Facilitate communication with stakeholders.</td>
<td>NPD-4; NPD-5</td>
</tr>
<tr>
<td>S-34</td>
<td>Test Driven Development</td>
<td>Write a failing automated test and develop code to pass the test</td>
<td>Clarify understanding of feature to be developed and improve design. Produce higher-quality code.</td>
<td>NPD-3; A-10</td>
</tr>
<tr>
<td>S-35</td>
<td>Refactoring</td>
<td>Improve code without changing behaviour.</td>
<td>Ensures incremental system growth without code decay. Facilitates code quality and maintenance.</td>
<td>NPD-4; A-9</td>
</tr>
<tr>
<td>S-36</td>
<td>Collective ownership</td>
<td>All developers empowered to work with any development artefacts, especially code and tests.</td>
<td>Enable knowledge sharing and continuous learning. Avoid specialization. Increased coverage of tasks.</td>
<td>NPD-2; NPD-4; A-9</td>
</tr>
<tr>
<td>S-37</td>
<td>Pair programming</td>
<td>Two developers work together in creating code.</td>
<td>Promotes collective ownership &amp; refactoring. Peer inspection increases quality. Facilitates knowledge sharing. Abstraction sharing helps design tests.</td>
<td>NPD-3; NPD-4; A-6; A-9</td>
</tr>
</tbody>
</table>

Table 3-7: Supplementary Scrum Practices - Technical practices (Cohen et al., 2004; Cohn, 2010)
3.3.4  Post-Sprint

The core practices of sprint review and sprint retrospective enable inspection and adaptation through reflection of both the work produced and the team performance. Insights uncovered from such reflection may lead to product backlog updates or modified team practices. Using a Scrum-master from another team to act as a facilitator for a Scrum retrospective may prove useful in disseminating information throughout the organization. A general technique for sprint retrospectives is to populate a two-column chart listing activities that ‘went well’ and ‘what could be better’. A variation on this technique is to annotate items with a ‘c’ if they were caused by Scrum and a ‘v’ if they were made visible by Scrum. This has the potential to reinforce Scrum as a useful technique for causing good work performance and exposing performance deficiencies (Deemer et al., 2010). The NPD principle of multi-level learning is enabled by this practice which is guided by the ASD principles of favouring face-to-face conversation in a regular reflection of the team’s performance.

An oft-cited core practice of Scrum is the ability to call the project done after any sprint i.e. every sprint produces a shippable product increment that is fully integrated with all prior increments. If this ideal is achieved, it alters the development mindset from a once-off project delivery style to continuous application or product development in a stable repetitive operational mode. Cohen et al. (2004) found that this approach leads to an emphasis by the team on system usability and correctness, rather than just strict feature growth. However, the ability to be done after any sprint is an ideal approach that may not always be possible due to weaknesses in an organization’s infrastructure and processes. In these cases, a release sprint may be necessary to ‘wrap up’ and coordinate release concerns. This activity forms part of the release management set of practices outlined in the next sub-section.
<table>
<thead>
<tr>
<th>Code</th>
<th>Practice</th>
<th>Description</th>
<th>Goal/Benefit</th>
<th>Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-38</td>
<td>Iterative controls / Sprint review</td>
<td>Broad concept of inspect and adapt on product / PO leads review: demo of potentially shippable code (max. 4 hours); Discuss product backlog and any changes to be made. Goal for next sprint defined. PO, SM &amp; team + any other stakeholders may ask questions.</td>
<td>Enable requirements elicitation and prioritization. Continuous improvement.</td>
<td>NPD-1; NPD-6; A-1; A-2; A-4; A-6; A-7</td>
</tr>
<tr>
<td>S-39</td>
<td>Iterative controls / Sprint retrospective</td>
<td>Broad concept of inspect and adapt on process / SM runs retrospective: reflection on team performance and possible improvements</td>
<td>Help prioritize requirements collection and design - meet customer’s needs. Facilitate risk management.</td>
<td>NPD-2; NPD-4; A-6; A-12</td>
</tr>
<tr>
<td>S-40</td>
<td>Ability to call the project done at any time</td>
<td>Can stop project for many reasons.</td>
<td>Every step treated as if it were the last iteration. Embed culture of usability rather than feature growth.</td>
<td>NPD-5; A-1; A-3</td>
</tr>
<tr>
<td>S-41</td>
<td>Sprint retrospective with cross-pollination</td>
<td>Team, p-owner and s-master + outside facilitator. (Use other s-master to achieve cross-pollination between teams). 2 charts: ‘working well’ &amp; ‘could be better’.</td>
<td>Continuous improvement</td>
<td>NPD-6; A-12</td>
</tr>
<tr>
<td>S-42</td>
<td>Retrospective cause analysis</td>
<td>‘c’ for caused by Scrum; ‘v’ for made visible by Scrum.</td>
<td>Continuous improvement</td>
<td>NPD-4; A-6; A-12</td>
</tr>
</tbody>
</table>

Table 3-8: Supplementary Scrum Practices - Post-Sprint (Cohen et al., 2004; Sutherland, 2006; Deemer et al. 2010)

### 3.3.5 Release Management

As stated above, in an ideal Scrum implementation, release planning is subsumed into the product-owner role and is a natural outcome of product backlog creation, grooming and sprint planning. However, there are cases where the incorporation of a release management role may prove useful. The release manager may work with the Scrum team to coordinate releases to customers and also to internally coordinate the staging of releases into a larger programme of work. Management of early adopter programmes is an example of an activity that may benefit from such a role. Monitoring release cadence of different Scrum teams can assist an organization in the determination of overheads associated with Scrum releases. This has the potential to improve release planning and control resulting in effective communication on release activities (Gannon, 2011).
<table>
<thead>
<tr>
<th>Code</th>
<th>Practice</th>
<th>Description</th>
<th>Goal/Benefit</th>
<th>Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-43</td>
<td>Release Sprint</td>
<td>Clean-up activities for release of product</td>
<td>Effective release management</td>
<td>NPD-6; A-1</td>
</tr>
<tr>
<td>S-44</td>
<td>Management reserve / contingency</td>
<td>Contract based fixed time/scope/date needs buffer for uncertainty</td>
<td>Project management</td>
<td>NPD-5</td>
</tr>
<tr>
<td>S-45</td>
<td>Release management role</td>
<td>‘Release manager’ who uses p-owner and developers to select appropriate customers to manage release. Addresses dependencies and communication issues with customer.</td>
<td>Effective release management</td>
<td>A-4</td>
</tr>
<tr>
<td>S-46</td>
<td>Release readiness reviews</td>
<td>Waterfall gate review concept to agree entry of Scrum release into larger product/program.</td>
<td>Effective release management</td>
<td>A-6</td>
</tr>
<tr>
<td>S-47</td>
<td>Release burndown chart</td>
<td>Analogous to Sprint burndown chart</td>
<td>Project management</td>
<td>NPD-2; A-8</td>
</tr>
<tr>
<td>S-48</td>
<td>Agile release communication</td>
<td>Monitor release cadence - inform of overhead associated with releasing shippable code at end of any particular sprint to facilitate release decision. Balance release cost, team velocity and released product utility.</td>
<td>Effective project management and release date prediction.</td>
<td>NPD-5; A-8</td>
</tr>
</tbody>
</table>

**Table 3-9: Supplementary Scrum Practices - Release (Cohen et al., 2004; Deemer et al., 2010; Gannon, 2011)**

Tables 3-4 to 3-9 presented a series of supplementary practices that may be used to enrich the core Scrum practice set. Each table proposed various NPD and ASD goals that may be achieved through performance of the practices. Table 3-10 presents an overview of these principles, presenting them across a summary of the reported variant or supplementary practices.
<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Scrum Practice</th>
<th>New Product Development</th>
<th>ASD Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-7</td>
<td>S-1 Rule of 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>S-2 Small teams</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>S-3 Feature teams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>S-4 Nurture self-organizing teams</td>
<td>√, √</td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>S-5 One project only</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>S-6 Whole-team response</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-2</td>
<td>S-7 One Product Owner</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>S-8 Merge roles</td>
<td>√, √, √</td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>S-9 Longevity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-8</td>
<td>S-10 Involve widely</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>S-11 Share Knowledge</td>
<td>√, √</td>
<td></td>
</tr>
<tr>
<td>CS-2</td>
<td>S-12 Testing debt</td>
<td>√</td>
<td>√, √</td>
</tr>
<tr>
<td>CS-2</td>
<td>S-13 Backlog estimates</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>S-14 Refinement</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-4</td>
<td>S-15 User stories</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-4</td>
<td>S-16 DEEP backlog</td>
<td>√, √, √</td>
<td></td>
</tr>
<tr>
<td>CS-2</td>
<td>S-17 Changes allowed</td>
<td>√, √, √</td>
<td></td>
</tr>
<tr>
<td>CS-4</td>
<td>S-18 Completeness</td>
<td>√, √</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>S-19 Consistent sprint</td>
<td>√, √</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>S-20 Mix stories</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>S-21 Groom backlog</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>S-22 Scrum board</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>S-23 Kanban</td>
<td>√, √, √</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>S-24 Protected sprint</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>S-25 Working software</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Core Scrum Practice</td>
<td>Scrum Practice</td>
<td>New Product Development</td>
<td>ASD Principles</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------</td>
<td>-------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>NPD-1 Built-in instability</td>
<td>A-8 Sustain Work</td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>NPD-2 Self-organizing Teams</td>
<td>A-9 Technical Excellence</td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>NPD-3 Overlap phases</td>
<td>A-10 Simple</td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>NPD-4 Multi-learning</td>
<td>A-11 Self-organizing Teams</td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>NPD-5 Subtle Control</td>
<td>A-12 Reflect &amp; Adjust</td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>NPD-6 Transfer of Learning</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-1 Satisfy customer</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-2 Welcome change</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-3 Frequent delivery</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-4 Engage Business Daily</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-5 Motivated Individuals</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-6 Face-to-face</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-7 Working Software</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-8 Sustain Work</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-9 Technical Excellence</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-10 Simple</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-11 Self-organizing Teams</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td>A-12 Reflect &amp; Adjust</td>
<td></td>
</tr>
</tbody>
</table>

Table 3-10: Summary of Scrum practices related to NPD and ASD principles

| CS-7                | S-26 Non-extendibility | √ | √ |                  |                  |
| CS-7                | S-27 Next sprint | √ |                  |                  |
| CS-7                | S-28 Sustainable pace | √ |                  |                  |
| CS-7                | S-29 Acceptance tests | √ |                  |                  |
| CS-7                | S-30 Frequent build | √ |                  |                  |
| CS-7                | S-31 Low-coupling | √ |                  |                  |
| CS-7                | S-32 Constant testing | √ |                  |                  |
| CS-7                | S-33 Constant doc. | √ | √ |                  |                  |
| CS-7                | S-34 Test-driven devt. | √ |                  |                  |
| CS-7                | S-35 Refactoring | √ |                |                  |
| CS-7                | S-36 Collectivism | √ | √ |                  |                  |
| CS-7                | S-37 Pair programming | √ | √ |                  |                  |
| CS-7                | S-38 Sprint review | √ | √ | √ | √ |                  |
| CS-7                | S-39 Sprint retrospective | √ | √ |                  |                  |
| CS-7                | S-40 Call done anytime | √ | √ | √ |                  |
| CS-7                | S-41 Cross-pollination | √ |                  |                  |
| CS-7                | S-42 Cause-analysis | √ | √ |                  |                  |
| CS-7                | S-43 Release sprint | √ | √ |                  |                  |
| CS-7                | S-44 Contingency | √ |                  |                  |
| CS-7                | S-45 Release manager |                  |                  |
| CS-7                | S-46 Release reviews |                  |                  |
| CS-7                | S-47 Release burn down | √ |                  |                  |
| CS-7                | S-48 Release commun. | √ |                  |                  |
3.4 Summary

This chapter opened with a description of the fundamental NPD principles that underpin the Scrum method (table 3-1) and a list outlining the ASD principles proposed in the agile manifesto (table 3-2). Scrum is introduced by presenting a set of standard practices gleaned from a review of authoritative publications on this framework (table 3-3). Following this introduction of the Scrum framework and its guiding principles, section 3.3 presented a number of supplementary Scrum practices that offer additional detail around variant practices that may be applied to perform the core Scrum practice set. Each supplementary Scrum practice is analysed and associated with appropriate NPD and ASD principles. A summary of how these principles are addressed across the various practices is presented in table 3-10. A synthesis of the supplementary practices into their associated core or standard Scrum practice reveals the various guiding principles associated with that practice.

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>New Product Development</th>
<th>ASD Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-2</td>
<td>Initial planning (and ongoing) - create or update product backlog</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CS-3</td>
<td>Pre-Sprint planning - create release backlog</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CS-4</td>
<td>Pre-Sprint planning - create sprint backlog</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CS-5</td>
<td>Pre-Sprint planning - define sprint goal</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CS-10</td>
<td>Post-Sprint - review meeting</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CS-11</td>
<td>Post-Sprint - team retrospective</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>CS-12</td>
<td>Definition of Done</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
</tr>
</tbody>
</table>

Table 3-11: Synthesis of guiding principles from supplementary practices to core set of practices

To carry this analysis further, these associated guiding principles could be considered potential goals that may be realized by applying the practice in different ways (variant practices). Table

---

5 Five principle associations indicated in **BOLD** are not identified in supplementary tables but have been added here.
3-11 presents the principles associated with each of the standard Scrum practices. Although not identified in any variant practices, certain principles are associated with some of the core Scrum practices by the author:

- The product owner’s ability to shield the team from distractions is identified as a form of ‘subtle control’ and as such is added to the above table for the core practice of Scrum team configuration.

- Sprint ‘goal definition’ (CS-5) is not explicitly identified in any of the supplementary practices reported. However, it is associated here with the NPD principle of ‘self-organizing teams’ and the ASD principle of A-10 ‘simplicity to maximize the amount of work not done’. The sprint goal is formulated by the team based upon the product owner’s needs and the backlog items selected for the sprint. It enables the team to both scope its immediate work and help avoid waste (e.g. design discussions related to non-sprint work).

- Similarly, these two principles are also associated with CS-12, ‘Definition of done’ by the author. It is deemed that this core Scrum practice enables the team to exhibit transcendence over time as they may increase the acceptance criteria for a ‘done’ item in order to set goals that challenge their development capabilities and promote performance improvement. (It should be noted that additional principles identified by the supplementary practice S-40 are also associated with CS-12).

The five tick marks representing associations not uncovered in the literature but proposed by the author are presented in bold type. As stated earlier, the core Scrum practice of ‘Sprint - Daily Scrum’ is expanded to consider ‘Scrum - general activities’. This is done to effectively synthesize the various principles associated with the different concepts associated with the notion of a Scrum iteration or sprint.

The chapter concludes by acknowledging that the Scrum framework has received special attention with regards to its application in different contexts. One such context is the use of the
framework in GSD settings. Practices associated with the application of Scrum in GSD are explored in detail in the next chapter.
Chapter 4- Application of Scrum in GSD

4.1 Introduction

Figure 4-1 presents the work packages that must be completed in order to extend basic Scrum concerns to its operation within GSD.

The purpose of this chapter is to investigate how Scrum may be used to alleviate issues that have been identified in relation to ISD conducted by globally distributed teams. The WBS segment shown in figure 4-1 above illustrates the three steps followed in this investigation. Section 4.1 presents a set of goals identified from literature that are pursued to alleviate or overcome GSD challenges. Reports on how core Scrum practices may help alleviate certain GSD concerns are introduced in section 4.2. The rest of this section reports on an analysis of GSD literature that uncovers a number of supplementary Scrum practices applied in a GSD context. Each of these practices is examined in order to propose which of the identified GSD challenges may be
alleviated through the application of the practice. Finally, section 4.3 presents a synthesis of the supplementary practices and GSD goal associations into the core practices proposed in chapter 3.

4.2 GSD Challenge Alleviation Goals

Successful implementation of GSD has to overcome various challenges. Ågerfalk et al. (2005) conducted a review of peer-reviewed literature on distributed development in order to propose a framework that presents the reported challenges and benefits encountered by teams performing ISD in this context. Although their work encompassed the broader context of distributed development, it reported on many findings related to globally distributed teams and their framework has been leveraged by subsequent publications on the application of ASD in GSD (Holmstrom et al., 2006; Hossain et al., 2011). The framework presented distributed development concerns across two dimensions: process and distance (table 4-1). The process dimension was explored using three widely reported processes in the context of GSD: communications, coordination and control.

(i) Communication involves the establishment of a common understanding through effective information transference between collaborating parties. It is a necessary process of all software development. However, its importance is magnified in distributed development due to the reduction in context caused by the substitution of face-to-face contact with technology solutions.

(ii) Coordination is concerned with the management of dependencies between people and tasks. One of its outputs may be the identification of the critical path of a set of tasks.

(iii) Control encompasses project management processes. It is concerned with project objectives in relation to duration, budget and quality.

References in the literature to the above processes were further investigated along the lines of three dimensions of distance: temporal (time), geographical (space) and culture.

(i) Temporal distance: This refers to the overlap in working hours between two distributed parties. An aspect of this distance relates to the concept of ‘follow the sun’. If there is
high temporal distance between sites, then there may be an opportunity to extend the working day if the tasks are suitable. However, this also means little or no overlap in time zones resulting in reduced opportunity for synchronous communication.

(ii) Geographical distance: The ease at which a party can physically visit a collaborating party’s office. This ease may be impacted by availability of transport infrastructure or political restrictions.

(iii) Socio-cultural distance: This measures the perception of one party about the values and practices of a remote party. This perception is influenced by a person’s organizational culture and their national culture. A low socio-cultural distance between parties would indicate that they perceive the other as having similar values and norms – this would assist their communication.

<table>
<thead>
<tr>
<th>Process</th>
<th>Dimension</th>
<th>Geographical Distance</th>
<th>Socio-Cultural Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Temporal Distance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time zone effectiveness</td>
<td>Proximity to market/customer</td>
<td>Innovation and shared best practices</td>
</tr>
<tr>
<td></td>
<td>Delayed communication</td>
<td>Lack of informal communication</td>
<td>Asynchronous communication preferred by non-native speakers</td>
</tr>
<tr>
<td></td>
<td>Delayed feedback</td>
<td>Dependency on ICT</td>
<td>Language differences and misunderstandings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Increased effort to initiate contact</td>
<td>Managing frames of reference</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Providing technical infrastructure</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cost of travel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Geographical Distance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access to large labour pool</td>
<td></td>
<td>Mix of skills and experiences</td>
</tr>
<tr>
<td></td>
<td>Standardisation in work practices</td>
<td></td>
<td>Language and cultural training</td>
</tr>
<tr>
<td></td>
<td>Allocation of roles and team structure</td>
<td></td>
<td>Lack of domain knowledge</td>
</tr>
<tr>
<td></td>
<td>Reduced trust</td>
<td></td>
<td>Doubtful of others’ capabilities</td>
</tr>
<tr>
<td></td>
<td>Lack of awareness/team spirit</td>
<td></td>
<td>Lack of mechanisms for creating shared understanding</td>
</tr>
<tr>
<td></td>
<td>Modularisation of work</td>
<td></td>
<td>Standardisation in work practices</td>
</tr>
<tr>
<td></td>
<td>Lack of mechanisms for creating shared understanding</td>
<td></td>
<td>Coordination complexity</td>
</tr>
<tr>
<td></td>
<td>Coordination complexity</td>
<td></td>
<td>Lack of awareness/team spirit</td>
</tr>
<tr>
<td></td>
<td>Socio-Cultural Distance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Management of project artefacts</td>
<td></td>
<td>Perceived threat from low-cost alternatives</td>
</tr>
<tr>
<td></td>
<td>Time zone effectiveness</td>
<td></td>
<td>Adapting to local formalized norm structures</td>
</tr>
<tr>
<td></td>
<td>Lack of concurrent engineering principles</td>
<td></td>
<td>Different perceptions of authority/hierarchy</td>
</tr>
<tr>
<td></td>
<td>Allocation of roles and team structure</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4-1: GSD Challenges and benefits (Ågerfalk et al., 2005)
Each cell in the GSD framework describes challenges and benefits arising from GSD cases. Emoticons are used to denote challenges (Ẽ) and benefits ( ☺). Open issues are tagged with a neutral symbol (균). Some issues are found in more than one cell. An issue is printed in standard typeface within the cell that is mainly responsible for its manifestation and other cells that influence that issue list it in italics.

<table>
<thead>
<tr>
<th>Process</th>
<th>Distance</th>
<th>Goal</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Communication</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temporal</td>
<td>com-t1: Overcome issues caused by reduced opportunities for synchronous communication</td>
<td></td>
<td>Delayed communication and feedback due to unavailability of resources and misunderstandings</td>
</tr>
<tr>
<td>Geographical</td>
<td>com-g1: Manage communication difficulties due to lack of easy access to other team members</td>
<td></td>
<td>Increased effort to initiate contact</td>
</tr>
<tr>
<td>Socio-cultural</td>
<td>com-s1: Avoid cultural misunderstandings that may lead to miscommunication</td>
<td></td>
<td>Misunderstandings caused by language or dialect differences</td>
</tr>
<tr>
<td><strong>Coordination</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temporal</td>
<td>coord-t1: Address increased coordination costs</td>
<td></td>
<td>Reduced overlapping hours for collaboration</td>
</tr>
<tr>
<td>Geographical</td>
<td>coord-g1: Overcome coordination difficulties due to lack of teamness</td>
<td></td>
<td>Difficulty in creation of trust from a spatial distance</td>
</tr>
<tr>
<td>Socio-cultural</td>
<td>coord-s1: Establish consistent work practices (or promote clear understanding of existence and rationale for different site work practices)</td>
<td></td>
<td>Sites may have doubts of technical competence of other sites</td>
</tr>
<tr>
<td>Socio-cultural</td>
<td>coord-s2: Avoid cultural misunderstandings that may lead to decreased cooperation</td>
<td></td>
<td>May need investment in language and cultural training</td>
</tr>
<tr>
<td>Temporal</td>
<td>con-t1: Overcome inconsistencies in management of artefacts</td>
<td></td>
<td>Ensuring process and artefact standards are maintained across sites</td>
</tr>
<tr>
<td>Geographical</td>
<td>con-g1: Difficulty implementing concurrent engineering principles</td>
<td></td>
<td>Need good tools and ability to manage change across space and time to ensure effective ISD</td>
</tr>
<tr>
<td>Socio-cultural</td>
<td>con-s1: Avoid morale issues by managing different perceptions of authority</td>
<td></td>
<td>Awareness of different cultural attitudes to authority</td>
</tr>
<tr>
<td>Socio-cultural</td>
<td>con-s2: Promote adaptation to local regulations</td>
<td></td>
<td>Visa applications, work-permits. Local development approaches</td>
</tr>
<tr>
<td>Socio-cultural</td>
<td>con-s3: Address misperception of threat from other sites</td>
<td></td>
<td>Need clear single point of accountability among sites to ensure cooperation</td>
</tr>
</tbody>
</table>

Table 4-2: Goals derived from GSD challenges (Ågerfalk et al., 2005)

The application of aspects of a method are dependent upon the method user’s beliefs and understanding of how those aspects will achieve certain goals for that user (Ågerfalk and Fitzgerald, 2006). Chapter 5 expands this concept further. In order for particular Scrum practices to be adopted within GSD, it is deemed necessary that they should be perceived by the user to
achieve certain goals in the context of a GSD project. Table 4-2 presents a set of goals related to the management of the GSD challenges summarized above. This table represents a subset of the larger group of goals to be addressed when performing GSD. The remainder of this section explores the application of Scrum in GSD to determine if Scrum is an appropriate vehicle for the achievement of this particular set of goals. Further opportunities for research include exploration into the efficacy of Scrum usage to achieve goals related to potential benefits associated with GSD. This is discussed further in chapter 10.

4.3 Application of Scrum practices to GSD

The Scrum roles of Product-owner and Scrum-master address potential communication issues at the boundary between disparate stakeholders. These ‘boundary spanners’ are complemented by the practices of User stories, Product backlog and Scrum board to effectively manage communication and coordination issues. Such practices are ‘boundary objects’. Clearly, Daily Scrum is also a powerful coordination vehicle, enabling teams to self-correct their immediate intentions based upon any reported obstacles or issues. The progressive elaboration of Scrum tasks from high-level user-stories on the product backlog through to more detailed descriptions and responsibility allocations as they are pulled into sprints enables successful ‘articulation’ and consequently effective communication. (From an LT perspective, this notion resembles the concept of addressing work at the ‘GEMBA’ or place of action). Team relationships or ‘social capital’ is strengthened by daily Scrum-of-Scrums and ‘all-hands’ sprint retrospectives. Trust issues provoked by socio-cultural distance may be alleviated using various Scrum techniques:

- Knowledge-based trust: ‘daily Scrum’ promotes interaction enabling team members to become more familiar with one another
- Identification trust: ‘sprint planning’ and ‘user-story prioritization’ can help team members develop this trust as they endorse each other’s values
- Performance trust: sprint burn-down chart and sprint reviews afford the visibility to teams of their competence and ability to meet commitments
The clarity of goals and results provided by punctuated development iterations (sprints) enables coordination among distributed team members (Pries-Heje and Pries-Heje, 2011).

Complementing the strengths provided by the core Scrum practices, a number of specialized practices have emerged to further enhance the usefulness of Scrum in managing GSD. Hossain et al. (2009) conducted a systematic literature review of Scrum practices applied in GSD. The aforementioned GSD framework (table 4-2) was used by Hossain et al. (2011) to synthesize 202 Scrum practices into eight ‘mitigation mechanisms’ that may be applied to address GSD challenges. A notable observation from their work is the confirmation that core Scrum practices can aid in overcoming GSD difficulties.

As stated earlier, Scrum is an empirical approach that requires adherence to its framework and associated core practices but does not prescribe additional specific development practices. As such, compilation of an exhaustive list of supplementary Scrum practices would be a highly difficult if not impossible task as different organizations may invent supplementary practices to meet a specific situational need. In order to suggest a practice set that may be used to alleviate GSD challenges, the goals identified in table 4-2 are aligned with particular Scrum practices acquired from various sources (Cohn, 2010; Dodda and Ansari, 2010; Cristal et al., 2008; Hossain et al., 2011). The following sections discuss supplementary Scrum practices that may be performed when using the framework in GSD. Using the same analysis structure presented in chapter 3, supplementary practices are described within four categories: ‘general considerations’ and different phases of a Scrum iteration. Tables containing supplementary practices are presented for each category. These tables describe each practice and propose associations between the practice and any GSD challenge goals that it may pursue. An additional GSD-related category presents the Scrum framework in the context of different distributed team configurations.

### 4.3.1 General considerations

Using information and communications technology (ICT) to support meetings is a general consideration that can help alleviate spatial distance. Video-conferencing software, supported by
groupware technology can help remote collaborators share applications or enrich the communication via facial expressions (Daft and Lengel, 1984). This can be extended to a dedicated ICT-enabled space which can be leveraged for informal communications thus increasing the chances of ‘opportunistic interactions’ (De Souza, 2001).

<table>
<thead>
<tr>
<th>ID</th>
<th>Practice</th>
<th>Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD-1</td>
<td>ICT supported meetings</td>
<td>Use ICT-mediated synchronous communication (e.g. Sametime). Requires solid telecommunications infrastructure</td>
<td>com-s1</td>
</tr>
<tr>
<td>GSD-2</td>
<td>Effective documentation management</td>
<td>Issue tracker to aid coordination. Configuration mgmt to integrate related documents. Support meetings/calls with written artefacts.</td>
<td>com-s1; coord-s2; con-t1</td>
</tr>
<tr>
<td>GSD-3</td>
<td>ICT-enabled dedicated workspace</td>
<td>Specific work space allotted to Scrum team. Variation is dedicated meeting room or breaks room with ‘always on’ videoconferencing. Encourage meetings/breaks to be held in these rooms. Enables informal communications between different locations</td>
<td>com-g1; coord-g1; coords-s2</td>
</tr>
<tr>
<td>GSD-4</td>
<td>Visits - sending &amp; maintaining</td>
<td>For various events/meetings, have distributed team members work face-to-face e.g. One sprint or a sprint-planning day. Useful for both building and maintaining relationships. Variation is to temporarily embed team members in different sites.</td>
<td>com-s1; coord-s1; con-s1</td>
</tr>
</tbody>
</table>

Table 4-3: General considerations - GSD Scrum practices (Hossain et al., 2011; Cohn, 2010; Dodda & Ansari, 2010)

Execution of the initial sprint in a collocated setting may help to build trust and move the distributed team through the initial team development stages of ‘forming’, ‘storming’ and ‘norming’ (Tuckman, 1965). Preparation and training of teams on the Scrum framework may help to overcome socio-cultural barriers. Recording the rationale for agreed process adjustments may also assist in this regard. Remote team members get an opportunity to understand the reasons that particular approaches to tasks are preferable to colleagues from a different background and compromises can be achieved and sustained. Establishment of the initial vision and product backlog can help the development team gain a common understanding of the key items to be produced. Increasing the detail on product backlog items to include conditions of satisfaction (or acceptance tests) can lead to greater clarity of the requirements and thus assist in subsequent coordination among teams.
### Table 4-4: Initial Planning - GSD Scrum practices (Hossain et al., 2011; Cohn, 2010; Dodda & Ansari, 2010)

<table>
<thead>
<tr>
<th>ID</th>
<th>Practice</th>
<th>Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD-5</td>
<td>Product backlog</td>
<td>Core practice - helps overcome teamness and promotes consistent mechanisms across sites</td>
<td>coord-g1; con-t1</td>
</tr>
<tr>
<td>GSD-6</td>
<td>GSD Scrum preparation</td>
<td>Ensure team members well resourced and trained to perform activities needed as part of distributed Scrum teams. Ensure rationale for process decisions recorded.</td>
<td>coord-s1</td>
</tr>
<tr>
<td>GSD-7</td>
<td>‘Team Gathering’:</td>
<td>Do initial iteration as co-located team in order to fully understand processes and interdependencies. (Also called ‘Sprint Zero’)</td>
<td>coord-s1</td>
</tr>
<tr>
<td>GSD-8</td>
<td>Increase product backlog detail</td>
<td>Support user story with description of test to indicate acceptance</td>
<td>com-t1; con-t1</td>
</tr>
<tr>
<td>GSD-9</td>
<td>Release Plan</td>
<td>Despite ideal of release-every-Sprint, many projects will plan release-specific packaging.</td>
<td>Coord-g1</td>
</tr>
</tbody>
</table>

4.3.2 *Sprint planning*

Involvement of the entire team in sprint planning enhances communication as team members understand the reason for performing tasks. The practice of sprint planning ensures that at an important stage of a project (commencement of each iteration), all team members are compelled to try and overcome temporal communication challenges. Various innovative practices such as product owner visits, equitable sharing of time-zones (promotion of fair rotation of non-regular work hours between locations) and recording of meetings contribute to greater communication and coordination among team members.

### Table 4-5: Pre-Sprint GSD Scrum practices (Hossain et al., 2011; Cohn, 2010; Dodda & Ansari, 2010)

<table>
<thead>
<tr>
<th>ID</th>
<th>Practice</th>
<th>Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD-10</td>
<td>Sprint Planning</td>
<td>Core practice - brings transparency to a project and encourages inclusiveness (variations below cover other GSD challenges)</td>
<td>coord-t1; coord-g1; coord-s1</td>
</tr>
<tr>
<td>GSD-11</td>
<td>‘Product Owner Visits’:</td>
<td>Product owner visits to reinforce product vision.</td>
<td>con-g1; con-s1</td>
</tr>
<tr>
<td>GSD-12</td>
<td>Sprint-planning: Phone management</td>
<td>Can be ‘Day long phone call’ with all team members to mimic collocated situation, or may be ‘Two phone calls’ over 2 days to facilitate temporal distance</td>
<td>com-t1</td>
</tr>
<tr>
<td>GSD-13</td>
<td>Sprint-planning: Delegated planning</td>
<td>Technical leads from different locations plan sprint.</td>
<td>com-t1</td>
</tr>
<tr>
<td>GSD-14</td>
<td>Recorded sprint-planning meeting</td>
<td>Enable playback by distributed teams to enhance understanding</td>
<td>com-s1; coord-t1; coord-s2</td>
</tr>
</tbody>
</table>
4.3.3 Sprint

The iterative inspect and adapt nature of Sprint and Daily Scrum encourages communication and coordination. Using a global ICT-supported task board helps coordination across space and time. Having a local Scrum team that coordinates with other teams via the Scrum of Scrums overcomes temporal communication issues. Promotion of lateral communication can overcome socio-cultural communication issues such as unwillingness to share bad news. If temporal issues are very challenging for some members of a team, daily Scrum practices may be modified. Dodda and Ansari (2010) noted daily Scrum variants such as local ‘mini-Scrum’ or email contributions. Other practices include the use of wikis to post updates in relation to daily Scrum items. This approach may be seen in situations where the majority of team members are collocated. However, Cohn (2010) recommended that care be taken with this option as it can dilute the commitment of the remote participants. Daily Scrum preparation including pre-meeting emailed contributions helps reduce meeting length. A shorter meeting length reduces the inconvenience of attendance for team members who must attend outside regular work-hours and helps maintain their commitment to daily Scrums.

Another practice that overcomes the control challenge of artefact management is called ‘transition backlog’. This proposes that remote team members can record ideas for improvement during the sprint, and their recorded improvements may be used as part of the agenda during the sprint retrospective (Hossain et al., 2011).

Table 3-6 presented a Sprint supplementary practice used to maintain the product backlog and ensure its readiness for future sprint planning: ‘grooming the product backlog’. It is recommended that this practice consume between 5 - 10% of a team’s time during their sprint (Deemer et al., 2010; Cohn, 2010). Although the ‘whole team responsibility’ ethos of Scrum would indicate use of the full team to participate in grooming or ‘pre-planning’ meetings, it could be very unwieldy for large widely distributed teams. A variation on this approach is to have a pre-planning team that accepts responsibility for grooming the backlog and communicating their work to the wider team. However, this exposes the team to a risk of reducing their sense of ‘teammness’ due to a reduced set of communication channels. This risk
may be mitigated somewhat by employing a balanced approach to pre-planning activities. Instead of confining the weekly product backlog meeting invitees to the pre-planning team, all other team members may also receive an option to attend if they wish. Furthermore, in the final week of the sprint, the grooming meeting could be made a mandatory meeting for the entire team. This reduces the meeting management and temporal distance overheads while permitting all the team to participate thus decreasing the chances of communication and ‘teamness’ dilution (Woodward et al., 2010).

<table>
<thead>
<tr>
<th>ID</th>
<th>Practice</th>
<th>Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD-15</td>
<td>Sprint</td>
<td>Core practice of Sprint enables the project management of overall development due to iterative inspect and adapt. Also core practice of daily Scrum encourages the initiation of communications between remote parties (Passivarra et al., 2009)</td>
<td>con-g1</td>
</tr>
<tr>
<td>GSD-16</td>
<td>Sprint using Daily Scrum</td>
<td>Daily Scrum encourages the initiation of communications between remote parties (Passivarra et al., 2009). Core practice of Simple planning performed to start Sprint and maintain work with daily Scrum increases ‘teamness’, reducing geographical coordination distance (Holmstrom et al., 2006). This requires ICT support outlined in practice variations below</td>
<td>coord-g1; coord-t1; coord-s2</td>
</tr>
<tr>
<td>GSD-17</td>
<td>Global task board</td>
<td>Common tool to plan and coordinate sprint activities</td>
<td>com-s1; coord-t1; coord-g1; coord-s2</td>
</tr>
<tr>
<td>GSD-18</td>
<td>‘SCRUM master needs to be a strong negotiator’</td>
<td>Scrum master needs to overcome distance issues and manage priorities so that requirements are met in agreed timeboxes</td>
<td>coord-g1</td>
</tr>
<tr>
<td>GSD-19</td>
<td>Synchronized working hours</td>
<td>Extended working hours; modified working hours</td>
<td>com-t1</td>
</tr>
<tr>
<td>GSD-20</td>
<td>‘Site based local SCRUM team’</td>
<td>Have local meetings - allow reps to coordinate work via Scrum of Scrums and promote lateral communication via additional specific purpose distributed meetings</td>
<td>com-t1</td>
</tr>
<tr>
<td>GSD-21</td>
<td>‘Modified SCRUM practices’</td>
<td>Mandatory 12 hour email turnaround on questions - limit team members attendance at Scrum</td>
<td>com-t1</td>
</tr>
<tr>
<td>GSD-22</td>
<td>‘Reduce SCRUM meeting length’</td>
<td>Well prepared meetings resulting in reduced meeting length e.g. Send prepared support materials</td>
<td>com-t1; com-s1</td>
</tr>
<tr>
<td>GSD-23</td>
<td>Single call daily Scrum</td>
<td>One call with all team members (mimic collocated)</td>
<td>com-t1</td>
</tr>
<tr>
<td>GSD-24</td>
<td>Written daily Scrum</td>
<td>Some members write their contribution. Attention needed to ensure issues are addressed</td>
<td>com-t1; com-s1</td>
</tr>
<tr>
<td>GSD-25</td>
<td>Regional daily Scrum</td>
<td>Temporal regions hold daily Scrum followed by trans-regional representatives meeting</td>
<td>com-t1</td>
</tr>
<tr>
<td>GSD-26</td>
<td>Alternate time of daily Scrum</td>
<td>Respect the normal working hours of different teams in order to overcome temporal distance</td>
<td>coord-s2</td>
</tr>
<tr>
<td>GSD-27</td>
<td>Everybody phone daily Scrum</td>
<td>Occasionally, mandate all members to call into daily Scrum (even if collocated). Empathise with others.</td>
<td>coord-s2</td>
</tr>
<tr>
<td>ID</td>
<td>Practice</td>
<td>Description</td>
<td>Goal</td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>GSD-28</td>
<td>Scrum of Scrum(s)</td>
<td>Used to coordinate multiple teams (as per Scaling Scrum). Easier to manage</td>
<td>coord-t1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>than daily Scrum due to reduced frequency and number of participants.</td>
<td></td>
</tr>
<tr>
<td>GSD-29</td>
<td>Transition backlog</td>
<td>Distributed member record and save ad-hoc improvement ideas for review in</td>
<td>con-t1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>retrospective.</td>
<td></td>
</tr>
<tr>
<td>GSD-30</td>
<td>‘Additional distributed meetings’</td>
<td>Informal meetings to discuss common concerns or work responsibilities.</td>
<td>coord-t1</td>
</tr>
<tr>
<td>GSD-31</td>
<td>Product backlog grooming meetings</td>
<td>Weekly meeting - 5-10% of Sprint. For distributed, use balanced team (Set of</td>
<td>coord-g1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>representatives usually. Last week of sprint all distributed team attend.</td>
<td></td>
</tr>
</tbody>
</table>

Table 4-6: Sprint GSD Practices (Holmstrom, 2006; Cristal et al., 2008; Hossain et al., 2011; Passivarra et al., 2009; Cohn, 2010; Dodda & Ansari, 2010; Woodward, 2010)

4.3.4 **Post-Sprint**

The Sprint review affords the distributed team a regular opportunity to reflect upon the product under development and their work performance. Coordination challenges are addressed as this mandatory review meeting exposes dependencies and issues around product features under development. It also enables team members to appreciate each other’s cultural backgrounds through shared experiences and challenges. Establishing a mandatory presentation in the sprint retrospective can ensure total involvement and overcome socio-cultural issues where team members may have differing attitudes toward authority and be unwilling to speak frankly about opportunities for improvement (Dodda and Ansari, 2010). In cases where a distributed team consists of a full Scrum team at each separate location, Woodward et al. (2010) recommended the execution of a ‘joint retrospective’ to supplement the individual retrospectives held by the separate Scrum teams. This helps promote ‘organizational dissemination of learning’ by encouraging appreciation of remote team practices. As stated in section 3.3.4, the ideal situation that each sprint should result in an immediately releasable product increment is very challenging and subject to the infrastructure and processes of the development organization. Woodward et al. (2010) endorsed this view and recommended that a release sprint could prove useful to package the system for release. Another potential output from such a sprint is the packaging of internal artefacts in order to support future maintenance activities. This may overcome the challenge of remote sites managing artefacts inconsistently. Finally, the practice of release retrospective reflects upon a longer time span than one sprint. This promotes reflection on the quality of a particular release and prompts consideration of team performance over time. This has the
potential to help distributed teams appreciate the potential effectiveness of diverse approaches to work.

<table>
<thead>
<tr>
<th>ID</th>
<th>Practice</th>
<th>Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD-32</td>
<td>Core practice of Sprint Review brings transparency. Prevents issues by frequent monitoring of output and clarification of requirements</td>
<td>com-s1; coord-g1; con-s1</td>
<td></td>
</tr>
<tr>
<td>GSD-33</td>
<td>‘Mandatory participation’: Each site location has to perform mandatory presentation as part of Sprint retrospective</td>
<td>con-s1</td>
<td></td>
</tr>
<tr>
<td>GSD-34</td>
<td>Retrospectives not optional, but.... Enable some leeway for team members to miss this mandatory meeting (with explanation)</td>
<td>com-t1</td>
<td></td>
</tr>
<tr>
<td>GSD-35</td>
<td>One location retrospective Occasional collocated retrospective to enable full and frank review of local performance.</td>
<td>com-g1</td>
<td></td>
</tr>
<tr>
<td>GSD-36</td>
<td>Joint Retrospective Where configuration is distributed Scrum of Scrums, hold joint retrospective following individual Scrum retrospectives</td>
<td>coord-g1; coord-s1</td>
<td></td>
</tr>
<tr>
<td>GSD-37</td>
<td>Release Sprint Wrap up productization and packaging issues</td>
<td>con-t1</td>
<td></td>
</tr>
<tr>
<td>GSD-38</td>
<td>Release Retrospective Retrospective spanning many sprints (or entire project)</td>
<td>coord-s1</td>
<td></td>
</tr>
</tbody>
</table>

Table 4-7: Post-sprint GSD Practices (Hossain et al., 2011; Cohn 2010; Dodda & Ansari 2010)

4.3.5 **Distributed Scrum team configurations**

Five ‘centripetal forces’ have been proposed as critical to effective GSD (Carmel, 1999). Four of these: ‘strong telecommunications infrastructure’, ‘collaborative technologies’, ‘clear methodology’ and ‘managerial techniques’ (such as visits to promote informal communication) are accounted for in the various practices listed in tables 4-3 to 4-7. The remaining force is the configuration of a team that is distributed across different locations.

Grinter et al. (1999) described a number of configuration structures and explained that certain structures may be more suited to particular development situations:

(i) Module-based: this structure is appropriate in product development where a cohesive modular product architecture facilitates a task allocation style that enables each distributed site be assigned responsibility for a specific module. Site-independence can reduce the need for inter-site coordination and associated informal communication.
(ii) Phase-based: teams responsible for different steps of the development process are assigned to different locations. This may be used to leverage specific knowledge (e.g. locating ‘requirements gathering’ near customers). However, project durations can suffer if a particular phase is delayed and cannot hand off development to the next phase. This approach requires clear definition of the phase to phase handover process.

(iii) Functional expertise: all resources with a particular expertise are collocated. This facilitates efficient use of resources, enabling load balancing of personnel so that projects may benefit from available expertise. It also promotes knowledge sharing of the site’s core domain. However, project coordination difficulties need to be considered.

(iv) Customization-based: this configuration separates core development from the work required to implement a product at a customer’s site. Close proximity to the customer supports collaboration and visibility to customer-specific requirements and helps establish clear division of product versions between ‘general available’ (GA) or open-market product and customer specific demands. However care must be taken to ensure sufficient support is made available to remote customization teams.

(v) Integrated or Time zone-based: this structure permits 24-hour development (often described as ‘Follow-the-sun’) (Grinter et al., 1999). At the end of a working day, incomplete tasks may be transferred to colleagues in another time zone although this technique is best followed in cases where task handovers require little preparation and have a low risk of misinterpretation (O’Conchuir et al., 2009).

Three distributed Scrum structures that are reported to be widely adopted are: ‘Isolated Scrums’, ‘Distributed Scrum of Scrums’ and ‘Totally Integrated Scrums’ (Sutherland et al., 2007).

(i) Isolated Scrums: This configuration involves one or more Scrum teams working with developer(s) from other locations. This is mainly used where offshore or outsourced locations may not be performing Scrum or may not contain cross-functional teams. Such a structure may occur within the ‘customization’, ‘function-based’ or ‘phase-based’ configurations outlined above. For example, an offshore performance engineering team may contain specific skills and roles needed to exclusively monitor and engineer
development work in order to address performance, reliability and product sizing considerations. Such a team may not use Scrum and may consist of technical engineers (‘function-based’ configuration). A concern with this configuration is that different development approaches may exacerbate socio-cultural communication issues.

(ii) **Distributed Scrum of Scrums:** In this configuration, cross-functional Scrum teams conduct Sprints within their own location and collaborate to manage a central product backlog. This is considered a best practice and is recommended for distributed Scrum development by the Agile Alliance. This configuration resembles the ‘module-based’ structure in that it strives to remove most dependencies between teams.

(iii) **Totally Integrated Scrums:** Scrum teams are cross-functional and team members are distributed across different locations. The application of this approach mimics collocation and provides ‘location transparency’. It is best suited to distributed teams that are very experienced in Scrum development.

Observation of a project that applied the ‘Totally integrated Scrums’ structure noted a specific centralization practice that is labelled by this thesis as: ‘Central Control’. All Scrum masters, architects and product owners were collocated leading to effective coordination of requirements prioritization and consistent Scrum team performance across the various geographies. This led to an additional role of ‘Chief Scrum Master’ to manage Scrum of Scrums. A practice for scaling Scrum, ‘Chief Product Owner’ was also vital in the context of distributed teams as it served the same purpose of effectively managing the total product vision and overall product backlog that was shared among the product owners of each of the individual teams (Sutherland et al., 2007). Practices that support different Scrum distribution configurations are outlined in table 4-8.
<table>
<thead>
<tr>
<th>ID</th>
<th>Practice</th>
<th>Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD-39</td>
<td>Isolated Scrum</td>
<td>Scrum team working with other teams that may not be cross-functional or may not even be agile.</td>
<td></td>
</tr>
<tr>
<td>GSD-40</td>
<td>Distributed Scrum of Scrums</td>
<td>Each location has its own cross-functional Scrum teams with clear interfaces. All Scrum teams integrated via a Scrum of Scrums</td>
<td>com-t1; coord-t1</td>
</tr>
<tr>
<td>GSD-41</td>
<td>Totally Integrated Scrum</td>
<td>Cross-functional Scrum teams have members distributed across locations</td>
<td>com-s1, coord-s2, con-s1</td>
</tr>
<tr>
<td>GSD-42</td>
<td>Central Control</td>
<td>Central location for Scrum-masters and product-owners</td>
<td>com-g1; coord-t1; coord-g1; con-g1; con-s1; con-s3</td>
</tr>
<tr>
<td>GSD-43</td>
<td>Chief Product Owner</td>
<td>Single point of accountability to manage overall product backlog</td>
<td>coord-g1; con-g1</td>
</tr>
<tr>
<td>GSD-44</td>
<td>Chief Scrum Master</td>
<td>Single point of accountability to represent Scrum-master issues</td>
<td>con-t1; coord-g1; con-s1; con-s2</td>
</tr>
</tbody>
</table>

Table 4-8: GSD Scrum team configuration Practices (Sutherland et al., 2007)

Many Scrum practices have been identified that help support ISD teams operating in GSD settings. The concepts of transparency, inspection and adaptation associated with the Scrum framework help to manage GSD challenges around communication, coordination and control. Table 4-9 summarizes all the practices listed earlier, identifying the particular challenges that are addressed by each practice. The left-most column of this table links the core Scrum practices presented in chapter 3, table 3-3 to each of the GSD Scrum variant practices.
<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>GSD Challenge</th>
<th>Communication</th>
<th>Coordination</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Core Scrum Practice**
  - CS-1: GSD-1 ICT support
  - CS-1: GSD-2 Doc. mgmt.
  - CS-1: GSD-3 ICT dedicated room
  - CS-1: GSD-4 Visits
  - CS-2: GSD-5 Product backlog
  - CS-1: GSD-6 GSD Scrum prep.
  - CS-1: GSD-7 Team gathering
  - CS-2: GSD-8 Backlog detail
  - CS-3: GSD-9 Release plan
  - CS-4: GSD-10 Sprint planning
  - CS-1: GSD-11 Owner visits
  - CS-4: GSD-12 Phone mgmt.
  - CS-4: GSD-13 Delegated planning
  - CS-4: GSD-14 Record meeting
  - CS-7: GSD-15 Sprint
  - CS-7: GSD-16 Daily Scrum
  - CS-9: GSD-17 Global task board
  - CS-7: GSD-18 Strong negotiator
  - CS-7: GSD-19 Sync working hours
  - CS-7: GSD-20 Local Scrum
  - CS-7: GSD-21 Modified Scrum
  - CS-7: GSD-22 Reduced length
  - CS-7: GSD-23 Single call Scrum
  - CS-7: GSD-24 Written Scrum

- **GSD Challenge**
  - Reduced synchronous work
  - Reduced Access
  - Misinterpret
  - Costs: Meeting attendance
  - Lack of 'teamness'
  - Inconsistent practices
  - Manage artefacts
  - Concurrent engineering
  - Authority; Regulations

- **Core Scrum Practice**
  - CS-1: GSD-1 ICT support
  - CS-1: GSD-2 Doc. mgmt.
  - CS-1: GSD-3 ICT dedicated room
  - CS-1: GSD-4 Visits
  - CS-2: GSD-5 Product backlog
  - CS-1: GSD-6 GSD Scrum prep.
  - CS-1: GSD-7 Team gathering
  - CS-2: GSD-8 Backlog detail
  - CS-3: GSD-9 Release plan
  - CS-4: GSD-10 Sprint planning
  - CS-1: GSD-11 Owner visits
  - CS-4: GSD-12 Phone mgmt.
  - CS-4: GSD-13 Delegated planning
  - CS-4: GSD-14 Record meeting
  - CS-7: GSD-15 Sprint
  - CS-7: GSD-16 Daily Scrum
  - CS-9: GSD-17 Global task board
  - CS-7: GSD-18 Strong negotiator
  - CS-7: GSD-19 Sync working hours
  - CS-7: GSD-20 Local Scrum
  - CS-7: GSD-21 Modified Scrum
  - CS-7: GSD-22 Reduced length
  - CS-7: GSD-23 Single call Scrum
  - CS-7: GSD-24 Written Scrum

- **Communication**
  - Geographical
  - Socio-Cultural

- **Coordination**
  - Temporal
  - Reduced
  - Synchronous work

- **Control**
  - Socio-Cultural
  - Authority; Regulations

- **GSD**
  - GSD-1
  - GSD-2
  - GSD-3
  - GSD-4
  - GSD-5
  - GSD-6
  - GSD-7
  - GSD-8
  - GSD-9
  - GSD-10
  - GSD-11
  - GSD-12
  - GSD-13
  - GSD-14
  - GSD-15
  - GSD-16
  - GSD-17
  - GSD-18
  - GSD-19
  - GSD-20
  - GSD-21
  - GSD-22
  - GSD-23
  - GSD-24

- **CS**
  - CS-1
  - CS-2
  - CS-3
  - CS-4
  - CS-5
  - CS-6
  - CS-7
  - CS-8
  - CS-9
  - CS-10
  - CS-11
  - CS-12
  - CS-13
  - CS-14
  - CS-15
  - CS-16
  - CS-17
  - CS-18
  - CS-19
  - CS-20
  - CS-21
  - CS-22
  - CS-23
  - CS-24
<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Communication</th>
<th>Coordination</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Temporal Reduced synchronous work</td>
<td>Geographical Reduced Access</td>
<td>Socio-Cultural Misinterpret</td>
</tr>
<tr>
<td>CS-7</td>
<td>GSD-25 Regional Scrum</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>GSD-26 Alternate time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>GSD-27 Everybody phone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>GSD-28 Scrum of Scrums</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>CS-7</td>
<td>GSD-29 Transition backlog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>GSD-30 Additional meetings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>GSD-31 Backlog grooming</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-10</td>
<td>GSD-32 Sprint review</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CS-11</td>
<td>GSD-33 Mandatory present</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-11</td>
<td>GSD-34 Retrospective leeway</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CS-11</td>
<td>GSD-35 One location retrosp.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-11</td>
<td>GSD-36 Joint Retrospective</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CS-12</td>
<td>GSD-37 Release sprint</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-11</td>
<td>GSD-38 Release retrospective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>GSD-39 Isolated Scrum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>GSD-40 Dist. Scrum of Scrums</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>GSD-41 Totally Integrated Scrum</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CS-1</td>
<td>GSD-42 Central control</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CS-1</td>
<td>GSD-43 Chief product owner</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CS-1</td>
<td>GSD-44 Chief Scrum master</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Table 4-9: Summary of Scrum practices related to GSD challenges
4.4 Summary

This chapter explored the application of Scrum in GSD settings. The challenges associated with GSD are explained and a series of Scrum practices that may be used to alleviate these challenges are described. A summary of this analysis is presented in table 4-9. Alleviation of GSD challenges may be considered goals of the variant Scrum practices. A synthesis of variant or supplementary practices into their standard or core practice (cf. table 3-3) permits the association of certain GSD challenge alleviation goals to particular standard Scrum practices.

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>CS-1</th>
<th>CS-2</th>
<th>CS-3</th>
<th>CS-4</th>
<th>CS-5</th>
<th>CS-6</th>
<th>CS-7</th>
<th>CS-8</th>
<th>CS-9</th>
<th>CS-10</th>
<th>CS-11</th>
<th>CS-12</th>
<th>CS-13</th>
<th>CS-14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum Team Configuration</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Initial planning (and ongoing) - create or update product</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Pre-Sprint planning - create release backlog</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Pre-Sprint planning - create sprint backlog</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Pre-Sprint planning - define sprint goal</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sprint - lock features for duration</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sprint - general activities</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sprint - technical activities</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Post-Sprint - review meeting</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Post-Sprint - team</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Definition of Done</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 4-10: Synthesis of goals that overcome GSD challenges from supplementary Scrum GSD practices to core set of Scrum practices

CS-1, ‘Scrum team configuration’ refers to the three core roles of product-owner, Scrum-master and team. The practices listed in table 4-8 present different mechanisms ranging from product manager hierarchies to various role and team distribution structures that alleviate many GSD challenges. The above table appears to indicate that no GSD challenge will be alleviated by using variations of CS-5 (Sprint Goal). However, it seems likely that the proposal and agreement of a common goal or vision for a sprint would act as mechanism to establish a common understanding of the project environment (coord-g1). The same holds true for the establishment

---

6 Challenge associations (in bold) for CS-5, CS-6 and CS-12 have been added by the author as no evidence of variant practices was found in literature on Scrum use in GSD settings.
of a clear definition of when a body of work is complete (CS-12). For this reason, the author has indicated that both of these practices may potentially achieve that goal (highlighted in bold). As stated in chapter 3, the core Scrum practice of daily Scrum was broadened to encompass general sprint management activities. However, most of the variants associated with this practice relate to the daily Scrum. The author has also proposed that locking the features to be developed in a sprint (CS-6) will help to reduce uncontrolled change thus overcoming issues with awareness of changes (com-g1) and tools needed to manage same (con-g1). No GSD challenge alleviation goals were identified in relation to the performance of Sprint-technical practices (CS-8). In a study on the impact of ASD to GSD, Ågerfalk (2006) notes that the extreme programming ASD technique of ‘pair programming’ reduced both temporal and socio-cultural distance within a particular GSD setting. Given this evidence, the impact of Sprint technical activities is explored further in the empirical research in order to see if the performance of certain technical practices within Scrum helps to overcome GSD challenges.

Both the successful pursuit of NPD and ASD principles, combined with the alleviation of GSD challenges are considered some of the goals to be achieved by users of the Scrum framework when it is applied in a GSD setting. This information may be used to leverage the method rationale analysis framework in order to further explore Scrum. Chapter 5 describes the origin of this framework and refers to previous research efforts that have used it to explore the relationship between ASD and GSD. This study extends such research by leveraging the values and goals identified in chapters 2, 3 and 4 in order to use the method rationale analysis framework to explore the relationship between LSD and ASD in a situation where an instance of ASD (Scrum) is applied in a GSD setting.
Chapter 5- LDScrum Framework

5.1 Introduction

Figure 5-1 details the portion of the overall project scope outlined in figure 1-7 that forms the work performed to complete the development of the conceptual model.

![Diagram of LDScrum Framework]

The purpose of this chapter is to use the work produced in the previous chapters to develop the candidate LDScrum model. This model forms the conceptual framework used to support subsequent empirical research. As seen in figure 5-1 above, the work described in this chapter involves the use of method rationale analysis theory to develop two relationships:

(i) The linkage between core Scrum practices and the distributed Scrum goals that they pursue (DScrum goal rationale)

(ii) The association between distributed Scrum goals and the LSD values that influence the desire to pursue them (LDScrum value rationale).
The initial topics addressed in this chapter focus on method rationale analysis theory. It begins by introducing ISD methods, method-tailoring and a disciplined approach to method engineering. The first section concludes with the presentation of a framework designed to incorporate method-engineering concepts with the behaviours of method users.

The chapter then proceeds to apply the framework to the topics addressed by this research. The two WBS work packages (1.4.2 and 1.4.3) highlighted in figure 5.1 above are covered in the remaining sections. Section 5.3 describes how the method rationale analysis framework is to be used as a lens to explore the relationship between LSD and Scrum in GSD (referred to for the remainder of the chapter as ‘distributed Scrum’ or DScrum). Two core entities within the framework are described in detail: LSD values and DScrum goals. A hierarchy of LSD values that anchor other LSD values is proposed. The goals analysed in chapters 3 and 4 are mapped to a set of distributed Scrum (DScrum) goals and a hierarchy is proposed showing DScrum goals that support the achievement of higher-order DScrum goals.

Section 5.4 extends our understanding of the relationship between LSD values and DScrum goals by presenting the LDScrum value rationale associating LSD values with DScrum goals. Finally section 5.5 selects three sets of core Scrum practices (or Scrum ‘method fragments’) and leverages the work of chapters 3 and 4 to present associations between core Scrum practices and DScrum goals. These proposed connections imply that a relationship exists between certain practices and particular LSD values (through the LDScrum value rationale association presented in section 5.4). These three practice-goal-value models form the candidate LDScrum model and are available to support empirical research into the relationship between LSD (values) and Scrum (practices and goals) applied to GSD.

5.2 Method Rationale Analysis Framework

This section introduces ISD methods and the need for their adaptation to meet differing needs. A brief overview is provided of the forces that impact the performance of a method resulting in the emergence of a ‘method-in-action’. Two forces in particular are highlighted - the reasons or
rationale used to employ a method and the developers that use the method. The discipline of method engineering is introduced to explain some proposed approaches to the rigorous treatment of method adaptation in order to assemble methods from components (or method parts). This is followed by a discussion on an extension of method engineering concepts that incorporates the dimension of ‘method rationale’ (or reasons for method component selection). Finally, the concept of an actor is linked with method rationale in order to present a framework that “facilitates analyses of rationality resonance” (Ågerfalk, 2006, p. 2) between parties. This method rationale analysis framework enables each party to explore the reasons for their use of particular method components and determine if they have a shared understanding and belief as to why those components should be used.

5.2.1 ISD Methods

Fitzgerald et al. (2002) defined an information systems development (ISD) method as ‘a coherent and systematic approach’ which is based upon a particular philosophy of systems development. Development philosophy refers to the main concerns being addressed by the development organization e.g. maintainability, security, system reliability, end-user involvement in system construction, speed of development etc. Whereas Scrum is often presented as an agile ‘method’, LSD is sometimes referred to as a management philosophy rather than a specific ISD method (Cohen et al., 2004).

In their high-level review of software development method usage, Avison and Fitzgerald (2003) stated that methods were primarily used to attain standard systems, better processes and improved end product. Kan et al. (1994) noted that initial applications of computing were the domain of scientists and enabled great advances in many technical fields. However, as computing became more pervasive in general business environments, the lack of attention to user needs resulted in an association of software development with poor quality. The term ‘software crisis’ emerged to describe the perception that software systems were constantly behind schedule, over-budget and incorrectly scoped (Fitzgerald et al., 2002). An outcome of this was the application of phased approaches to development based upon the systems development life cycle. Fitzgerald et al. (2002) commented that the SDLC is not really a method in itself but
rather a higher-level concept that forms the basis for many methods. It demands that developers take a basic specify-design-build-test approach. Issues arising from this approach led to a proliferation of methods. Avison and Fitzgerald (2003) categorized these methods into seven broad ISD method categories: structured or process-flow based, data based, prototyping or representation based, object-oriented, participative (user-based), strategic and system-based. Many blended approaches also emerged, and tools were created to support the variety of available methods (Wistrand, 2009). These ranged from CASE tools to project management applications. Commercial methods were either used as purchased or tailored to meet the needs of individual organisations. When considering methods and method use, it is important to appreciate the distinction between a formalised or defined method and its application in a development environment. The general nature of many methods dilutes their effectiveness to meet the needs of ‘real’ projects and necessitates consideration on how best to either reduce or augment methods to meet the needs of particular projects (Brinkkemper et al., 1998). When adapted for different development situations, a formalized method becomes a ‘situational method’. Furthermore, when this situational method is used by a development team, it is tailored or adapted by the team to become their ‘method-in-action’ (Ågerfalk and Fitzgerald, 2006).

5.2.2 Method-in-action

Figure 5-2: Framework for ISD method use (Fitzgerald et al., 2002)
The method-in-action is influenced by many forces including available formalized methods, development context, the type of system under development, reasons for selection or use of the method and developer experience. The latter two forces are of particular interest to this thesis as they relate clearly to the ultimate framework used for data collection and analysis.

Method selection and adaptation by ISD groups may be influenced by both rational and political motivations. The decomposition of a software development initiative into pre-defined steps carried out by different players from the development community fuels the rational role of a method. It enables clear project management and effective allocation of heterogeneous resources to different development phases. On the other hand, a method may also play a political role: it may be a ‘crutch’ that allows a development group to justify their performance. Methods may also be used to demonstrate the complexity of professional software development, and thus protect against unreasonable client demands (Fitzgerald et al., 2002).

Developers have varying skill levels — both in technical competency and also in their domain knowledge. The need for composition or abstraction skills in software development adds greatly to the difficulty of the work. How do developers ‘imagine’ the system to be created? How do they effectively communicate an intangible vision of a potential system? Due to the complexity of the ISD process it is important that developers exhibit both communication and pedagogical skills. The ability to abstract a potential system and then communicate this effectively is a difficult skill to master. Indeed, the creation of new information systems often results from the learning process undergone by developers and users as they see the future system emerging. Therefore, this challenges the idea of a simple formal method that outlines a series of orderly phases resulting in a new system. Methods need to embrace the learning that occurs from trial and error. This is in keeping with the idea that ideal formalised methods are actually tailored to become methods-in-action when put into use. A relationship exists between developer experience and method usage (figure 5-3). Essentially, inexperienced developers, regardless of their role in the development (business analysts, designers, programmers, testers, users…), rely heavily on the method that they have been taught. For example, before performing an activity,
they may look up the online system or template that supports their development method and then blindly adhere to the instructions in it. As they gain experience, an element of cynicism may emerge. They may find that many demands of the formalised method do not add value, and are merely bureaucratic form-filling exercises. With more experience, this cynicism gives way to insightful reflection of useful aspects of the formalised methods. This leads to increased method usage, but in a manner that incorporates tailoring of the value-add parts of the method (Fitzgerald et al., 2002).

Figure 5-3: ISD method used and developer experience (Fitzgerald et al., 2002)

5.2.3 Method Assembly

The notion presented by Goldkuhl et al., (1998) of a method consisting of a process, concepts and a notation (figure 5-4) is captured in the definition of all or part of a development method as a ‘method fragment’ (Brinkkemper et al., 1998). The concept of ‘method fragment’ is proposed
as a means to assemble methods from different components or fragments in order to build situational methods as needed.

In order to perform effective hierarchical method modelling (meta-modelling), a framework is introduced that presents method fragments as having three dimensions that should be considered when determining how to tailor or construct a situational method: perspective, abstraction level and layer of granularity (table 5-1). The perspective of a method fragment refers to either the product created by the fragment or the process used to implement it. Abstraction level addresses either conceptual method descriptions or technical tools used to implement the method (e.g. CASE tool). The granularity dimension classifies the level of the method fragment (Brinkkemper et al., 1998).

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Instance</th>
<th>Description of part of method</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perspective</td>
<td>Product</td>
<td>Deliverables, models, diagrams etc.</td>
<td>State chart</td>
</tr>
<tr>
<td></td>
<td>Process</td>
<td>Stages, activities, tasks</td>
<td>State chart construction</td>
</tr>
<tr>
<td>Abstraction Level</td>
<td>Conceptual</td>
<td>Description of part of method</td>
<td>A state chart is....</td>
</tr>
<tr>
<td></td>
<td>Technical</td>
<td>Tool(s) to support conceptual description</td>
<td>Rational Rose</td>
</tr>
<tr>
<td>Granularity</td>
<td>Method</td>
<td>Complete method</td>
<td>Rational Unified Process</td>
</tr>
<tr>
<td></td>
<td>Stage</td>
<td>Addresses a segment of the ISD Lifecycle</td>
<td>Requirements Spec.</td>
</tr>
<tr>
<td></td>
<td>Model</td>
<td>Perspective of IS (e.g. Aspect of abstraction)</td>
<td>Data model or UI model</td>
</tr>
<tr>
<td></td>
<td>Diagram</td>
<td>Representation of a view of model layer fragment</td>
<td>State diagram (chart)</td>
</tr>
<tr>
<td></td>
<td>Concept</td>
<td>Atomic building blocks (process &amp; notation)</td>
<td>Associations, objects</td>
</tr>
</tbody>
</table>

Table 5-1: Dimensions of method fragments used in situational method assembly (Brinkkemper et al., 1998)
The application of this framework and associated constraints and rules can simplify and guide the assembly of methods from method fragments. For example, in certain cases method fragments must only be assembled from the combination of fragments at the same classification (perspective, level and granularity). Use of method fragments and method assembly rules guide situational method engineering (Brinkkemper et al., 1998).

This research study leverages the method fragment structure by introducing three method fragments that are combined to form the entire Scrum method. In relation to the three dimensions described above, these method fragments are proposed from a process perspective and at the conceptual abstraction level. They represent three sub-sets of core Scrum practices. These method fragments are presented at the stage level of granularity as they represent different phases in the ISD lifecycle - ‘pre-sprint’; ‘sprint’ and ‘post-sprint’ practices. It is not the intention of this study to pursue the engineering of these method fragments further. The high-level method assembly concepts outlined above are presented for background informational purposes only in order to provide some context around the origins of the method fragment structure. This structure is a core component of the LDScrum conceptual framework and is used to facilitate the application of method rationale theory to portions of the Scrum framework.

5.2.4 Method Rationale

An additional dimension of a fragment is its rationale - the intention or goal of the method fragment user and the value that underpins that goal. This indicates that every method fragment is performed to achieve at least one or more goals and comply with one or more values associated with the philosophy of the overall method (Ågerfalk and Wistrand, 2003).

Every method must have at least one method fragment. Performance of this fragment should lead to the achievement of one or more goals. These goals have been established by the creator of the method and reflect their attitudes toward ISD i.e. each goal is anchored by one or more values. Figure 5-5 illustrates that method rationale consists of two sub-rationales:
(i) Goal rationale: method fragments exist to achieve one or more goals (and a goal may be achieved by multiple method fragments)

(ii) Value rationale: A goal is anchored by one or more values (and one value may anchor multiple goals).

Figure 5-5: Method Rationale (Ågerfalk and Wistrand, 2003)

Additionally, the figure suggests that a goal hierarchy may exist where certain goals may support the achievement of higher-order goals and a value hierarchy may exist where particular values may be anchored by higher-order values. It is also possible that goals may contradict other goals and values may contradict other values. Ultimately, this indicates that any method fragment present in a method should be there to achieve at least one goal and fulfil one or more of the values associated with the philosophy of the method (Ågerfalk and Wistrand, 2003).

5.2.5 Method Rationale Analysis Framework

The transformation of an ideal typical method constructed in a laboratory into a situational method by a process engineer is influenced by the values, beliefs and understanding of both method creator and adapter. Subsequently the manner in which the method is used by the developer is also influenced by their interpretation of the situational method and this interpretation is influenced by the values, beliefs and understandings of the developer. The
values and goals that prompt the use of various method fragments are leveraged by the method creator. However, as stated above, before use, an ‘ideal typical method’ is often tailored. Method rationale refers to the reasoning applied to the adaptation of a method by users. It may be considered from the viewpoint of design rationale where decisions taken in the method creation assist in understanding the method. Alternatively, method rationale may consider goals and values as described earlier to understand the reasons for the presence of method fragments. Methods may exist at three different levels (figure 5-6).

Figure 5-6: Levels of method abstraction (Ågerfalk & Fitzgerald, 2006)

A method creator develops an ‘ideal typical method’. This general ‘one size fits all’ method is then configured to suit a particular situation (domain or company). Finally, the developers applying the configured method will further adapt it to address their specific needs. Whereas the ideal typical and configured methods are documented as linguistic expressions of knowledge, the ‘method-in-action’ is expressed as action knowledge (which may or may not be documented linguistically). A key consideration in this continuum is that both the ideal typical and configured methods are suggestions of how ISD should be conducted and are produced from the subjective knowledge of the method creator or configurator. However, whether the suggestions are effectively interpreted by the method receiver is dependent upon the inter-subjective knowledge of producer and receiver i.e. the effective configuration of an ideal typical method is dependent upon the shared understandings, beliefs and values of the creator and configurator (or process engineer). Similarly, the inter-subjective knowledge between the configurator and developer will determine how the method-in-action is adapted from the situational method. Analysis of the ideal typical method fragments to reveal the goals and underlying values of the method creator can
explicate the method rationale and assist in developing shared understandings that may enhance
the adaptation process (Ågerfalk and Fitzgerald, 2006).

This indicates that the analysis of a method fragment using method rationale must also consider
the subject or actor involved in creating, adapting or reviewing a method. Ågerfalk (2006)
proposed a ‘method rationale analysis framework’ (figure 5-7) that links method fragments to
user goals in order to uncover the values that motivate method use. This framework may then be
used to investigate the “rationality resonance” (Ågerfalk, 2006, p.2) between method creators
and interpreters (i.e. the level of consistency in the method rationale or inter-subjective
knowledge between both parties). Rationality resonance is achieved when two or more actors
select a method fragment in order to achieve the same goal driven by the desire to address the
same value.

Figure 5-7: Method Rationale Analysis Framework (Ågerfalk, 2006)

The framework (figure 5-7) extends method rationale (figure 5-5) by adding the actor entity to
the model, thus necessitating the inclusion of two additional entities related to the actor:
‘method-in-action’ and ‘value base’.
Method-in-Action. As stated earlier, a method fragment may be an entire method or a part of a method (e.g. a practice or set of practices). However, the method rationale analysis framework states that not only may the method fragment be documented or linguistically expressed (‘in-concept’), it may also be performed by the actor (‘in-action’). It proposes that one aspect of rationality resonance, goal rationale, is achieved when two actors interpret the same method fragment as the means to achieve a particular goal. It could be useful for actors such as a method configurator and a method user (developer) to achieve rationality resonance as this would assist in the consistent application of a method fragment as it was intended to be used by the configurator. Alternatively, actors could be multiple developers. Exploring the goal rationale resonance between these parties could expand their understanding on the efficacy of method fragments, especially if the presence or lack of resonance is observed from the application of such fragments in development situations (‘in-action’).

Value base. Furthermore, the framework indicates that values guide the selection of goals to be achieved. Value rationale is evident when multiple actors seek to achieve the same goal driven by the need to address the same value(s). An actor’s values reside in their value base. It should be noted that the framework suggests that an actor’s value base may be empty as it is possible for values to emerge in an individual as they progress with their work.

One particular application of the method rationale analysis framework explores the relationship between ASD and the application of one particular ASD method (XP) in a GSD environment. ASD values and XP goals are used to determine the consistency between certain XP method fragments (or practices) ‘in-concept’ and their application ‘in-action’ (Ågerfalk, 2006). A recommendation arising from this research was that a more in-depth analysis of agile rationale should be undertaken and insights arising from such work should be used to inform the execution of more in-depth case studies on agile methods.
This thesis seeks to use the method rationale analysis framework to explore a particular subset of agile rationale. It is used as a lens to examine the relationship between LSD and one particular manifestation of ASD: Scrum in a GSD setting. The outputs from earlier chapters provide the necessary data to inform an initial linkage between these ISD concepts. Chapter 2 analysed the literature in order to derive a set of LSD values and associated principles (or goals). Chapters 3 and 4 explored the rationale for the implementation of many Scrum practices by linking them to particular goals. The remainder of this chapter describes the construction of various parts of the candidate LDScrum model. This model is an instance of the method rationale analysis framework. In accordance with the recommendations of (Ågerfalk, 2006), this model will be used to execute an in-depth case study on a specific agile implementation. Section 5.3 presents the potential LSD value base and distributed Scrum (DScrum) goal hierarchy. Section 5.4 describes a potential value rationale that links LSD values to distributed Scrum (DScrum) goals. Section 5.5 selects particular sets of Scrum method fragments and proposes a goal rationale for their use. The presentation of each method rationale linking particular DScrum method fragments to goals and associated LSD values is considered from the viewpoint of an ISD developer on a distributed Scrum team.

5.3 Application of framework - LSD value base & DScrum goal hierarchy

The previous section explained that the method rationale analysis framework describes the goal and value rationales used by an actor in the performance of a method fragment (figure 5-7). In the context of this study, the method fragments being explored are three sets of core Scrum practices representing pre-sprint, sprint and post-sprint phases of the method. The candidate LDScrum model represents the application of this framework to these three method fragments in order to propose an ‘ideal’ description of the rationale. To this end, the actor is deemed to be an ISD developer performing Scrum in a distributed team. The term ‘developer’ refers to a member of a professional software development team and as such encompasses various roles including project manager, software architect, software designer, coder, quality assurance specialist etc. The amalgamation of multiple specific roles under the umbrella term ‘developer’ is an approach
adopted in earlier research into ISD methods (Fitzgerald et al., 2002). Clearly, developers will contain their own value base depending upon their background and experience. However, as stated earlier, the candidate LDScrum model is a conceptual framework intended for use as a lens to explore the relationship between LSD and the application of Scrum in GSD. As such, it will be constructed to analyse an ‘in-concept’ method fragment performed by an ‘ideal’ actor who may hold all possible LSD values within its value base. The refined LDScrum model to be constructed from empirical research will reflect the views of ‘real’ developers and as such will reflect ‘in-action’ instances of these three method fragments. Section 5.3.1 below explains how the LSD value base was constructed from LSD values identified in chapter 2 (table 2-3). Section 5.3.2 explains the DScrum goal hierarchy containing goals identified in chapters 3 and 4 (cf. table 5-2). Goals from this hierarchy form candidate intentions that motivate the selection of Scrum practices.

5.3.1 LSD Value Base

Chapter 2 identified principles reported in literature on the application of LT in the domains of operations management, product development, project management and software development. These principles are then analysed in order to propose a set of LSD values (table 2-3). The output of chapter 2 is a set of LSD values and the LSD principles associated with each value.

Values may be described as ‘guiding stars’ or action rules that enable individuals to make decisions (Senge, 1990). They reflect general beliefs and understandings that may be applied to particular situations. Values may be considered as more abstract than principles or goals. Value rationale refers to the reasoning applied by an actor to seek the achievement of certain goals in order to address particular values. An actor leverages a value base in order to determine which goals are worth pursuing. Certain values may be present in order to comply with higher-order values. It is possible that an actor may hold contradictory values. It is possible that the value base may be empty and that realization of certain values may only emerge as an actor is confronted with particular decisions or problems. (Ågerfalk, 2006).
Synthesis of the LSD values (figure 5-8) reveals an interesting dichotomy of two overarching goals that address an external and internal focus respectively: delivery of customer-defined value and reduction of waste. This focus on balancing the management of both external and internal issues is consistent with the duality observed in chapter 1 with respect to LT, as applied within LM, SCM and LSD.

![Figure 5-8: Potential LSD Value base](image)

The identification, development and delivery of customer-defined value reflects a core value that influences engagement with market forces. A ‘sub-value’ that influences a particular aspect of this management of customer needs is ‘Product excellence’. This involves building a culture of excellence that emphasizes design-oriented processes in order to avoid premature agreement on solutions and to promote the development of change-tolerant flexible systems. As such, other externally-oriented values that contribute to this culture are ‘Embrace change’ and ‘Business
environment awareness’. Finally, an internally-oriented value, ‘Effective process’ helps developers and teams to realize the value of product excellence. Another internally-oriented value, ‘Flow of value’ may also be considered a specific contributor to the realization of the value of addressing customer-defined value. Flow of value encourages pull which promotes continuous awareness of a customer’s perception of value. Rapid delivery of a customer request facilitates early verification that it has provided the value perceived by the customer. If this is not the case, then flow of value eases the development and delivery of a modification to the request in order to deliver the customer’s modified perception of value.

The highest-order internally-oriented value in LSD addresses the need for waste reduction. This is supported by the sub-value ‘Flow of value’. The promotion of demand-driven development assists with overall waste reduction as it avoids the provision of un-needed functionality and exposes bottlenecks. The rapid resolution of such bottlenecks is encouraged in order to resume flow. ‘Continuous improvement’ promotes waste reduction as learning is achieved and disseminated among the team and organization. ‘Person-focus’ encourages effective collaboration and the empowerment of developers to manage their tasks which may lead to the exposure and removal of many wasteful activities (such as the production of unused artefacts).

The set of externally-oriented LSD values $V_E$ outlined above is: \{V_1: Customer Value, V_6: Product Excellence; V_7: Embrace Change; V_8: Business Environment Awareness\}. The set of internally-oriented values $V_I = \{V_2: Reduce Waste, V_3: Flow of Value, V_4: Person Focus, V_5: Continuous Improvement, V_9: Data-driven Decisions, V_{10}: Effective Process, V_{11}: Effective Collaboration, V_{12}: Effective Use of Technology\}. The overall set of LSD values $V = V_E \cup V_I$.

The value anchoring relation (described as ‘is anchored by’), $VAR \subseteq V \times V =$

$$\{ (v_{12}, v_{10}), (v_{10}, v_3), (v_{10}, v_9), (v_{12}, v_4), (v_9, v_2), (v_3, v_1), (v_3, v_2), (v_4, v_2), (v_5, v_8), (v_7, v_9), (v_9, v_8), (v_{10}, v_7), (v_9, v_7)\}.$$

The value contradiction $VCR \subseteq V \times V = \emptyset.$
5.3.2 Distributed Scrum Goal Hierarchy

The exploration of Scrum in chapters 3 and 4 leveraged both NPD characteristics and ASD principles as guiding principles (or goals) for this method. Guiding principles may be viewed as goals because the successful pursuit of a guiding principle may be considered the achievement of a goal. A core set of Scrum practices are identified. Analysis of the literature uncovered a series of supplementary practices that may be used to supplement core practices. Each supplementary practice was analysed in order to determine the guiding principle(s) that informed its use. Each of the core set of Scrum practices were then aligned with guiding principles that may be followed (or goals that may be achieved) by using particular supplementary practices that support that practice. Furthermore, the application of Scrum in GSD was explored and analysed. This analysis was facilitated by the determination of a number of goals that enable teams to overcome GSD challenges that have been reported in the literature. Supplementary Scrum practices performed by GSD teams were analysed to determine what GSD challenge alleviation goals inform their use. Finally, each core Scrum practice was aligned with the relevant GSD challenge alleviation goal(s) that may be pursued by using a supplementary practice associated with it. The outcome of chapters 3 and 4 was a core set of Scrum practices that are aligned with the NPD, ASD and GSD goals that may be achieved by their use (tables 3-11 and 4-10).

Table 5-2 combines the aforementioned NPD/ASD principles and GSD challenge alleviation goals into a single set of Distributed Scrum (DScrum) goals. These goals are used to adapt and extend earlier analysis of ASD goals by Ågerfalk (2006) in order to incorporate the NPD and GSD goals into a goal hierarchy (cf. figure 5-10). Goal rationale addresses the identification of goals that inform the selection and use of methods (either in-concept or in-action). A secondary consideration here is that the pursuit of certain goals may be motivated by the perception of them as contributing to the achievement of ‘higher-order goals’ (Ågerfalk, 2006).
<table>
<thead>
<tr>
<th>Goal</th>
<th>Code</th>
<th>Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>g1</td>
<td>NPD-1</td>
<td>Built-in instability</td>
</tr>
<tr>
<td>g2</td>
<td>NPD-2</td>
<td>Self-organizing project teams</td>
</tr>
<tr>
<td>g3</td>
<td>NPD-3</td>
<td>Effectively Overlap development phases</td>
</tr>
<tr>
<td>g4</td>
<td>NPD-4</td>
<td>Consistent Multi-learning</td>
</tr>
<tr>
<td>g5</td>
<td>NPD-5</td>
<td>Leverage Subtle control over work tasks</td>
</tr>
<tr>
<td>g6</td>
<td>NPD-6</td>
<td>Effectively disseminate learning throughout Organization</td>
</tr>
<tr>
<td>g7</td>
<td>A-1</td>
<td>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</td>
</tr>
<tr>
<td>g8</td>
<td>A-2</td>
<td>Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.</td>
</tr>
<tr>
<td>g9</td>
<td>A-3</td>
<td>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
</tr>
<tr>
<td>g10</td>
<td>A-4</td>
<td>Business people and developers must work together daily throughout the project.</td>
</tr>
<tr>
<td>g11</td>
<td>A-5</td>
<td>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</td>
</tr>
<tr>
<td>g12</td>
<td>A-6</td>
<td>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</td>
</tr>
<tr>
<td>g13</td>
<td>A-7</td>
<td>Working software is the primary measure of progress.</td>
</tr>
<tr>
<td>g14</td>
<td>A-8</td>
<td>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</td>
</tr>
<tr>
<td>g15</td>
<td>A-9</td>
<td>Continuous attention to technical excellence and good design enhances agility.</td>
</tr>
<tr>
<td>g16</td>
<td>A-10</td>
<td>Simplicity – the art of maximizing the amount of work not done – is essential.</td>
</tr>
<tr>
<td>g17</td>
<td>A-11</td>
<td>The best architectures, requirements, and designs emerge from self-organizing teams.</td>
</tr>
<tr>
<td>g18</td>
<td>A-12</td>
<td>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.</td>
</tr>
<tr>
<td>g19</td>
<td>GSD-1</td>
<td>Overcome GSD communication challenges (com-t1; com-g1; com-s1)</td>
</tr>
<tr>
<td>g20</td>
<td>GSD-2</td>
<td>Overcome GSD coordination challenges (coord-t2; coord-g2; coord-s2)</td>
</tr>
<tr>
<td>g21</td>
<td>GSD-3</td>
<td>Overcome GSD control challenges (con-t3; con-g3; con-s3)</td>
</tr>
</tbody>
</table>

Table 5-2: DScrum - Goal to principle mapping

ASD principles reflect aspirations on how best to address both external and internal concerns. Successful pursuit of particular ASD principles contributes to the goal of achieving other principles. Classification of ASD principles and the establishment of hierarchical relationships between principles may be presented as ‘goal clusters’ that exhibit either an external or internal orientation (Ågerfalk, 2006). Figure 5-9 presents an adaptation of the original goal clusters proposed by Ågerfalk (2006) to reflect the DScrum goal codes assigned in table 5-2. The presence of a ‘+’ on a link denotes that the lower goal supports achievement of the higher goal.
whereas a ‘-' sign indicates a contradiction between goals. Note that the adaptation of the goal codes from the original work is not a simple mapping resulting in a numerical shift upward by 6 for each code i.e. A1 - A12 to be changed to G7 - G18. This is due to some alterations made to the principles in the original analysis: decomposition of the first ASD principle (A-1) into two goals (satisfy customer and frequent delivery) and the omission of the sixth principle (face-to-face communication). For example, in the figure below, two of the externally-oriented goals are described by the same goal-code (g-7).

![Figure 5-9: ASD goal clusters (Ågerfalk, 2006) - goal codes adapted to reflect DScrum goal identifiers from table 5-2](image)

Commentators on different aspects of LT also consider this subject from the viewpoint of its focus on external and internal issues. Womack and Jones (2003) presented LT as a paradigm that addresses principles concerned with external concerns (customer-defined value) and internal concerns (removal of waste and continuous improvement). LT application in operations is
described as being both strategic (external focus on the comprehension and delivery of customer-defined value) and operational (internal shop-floor focus) (Hines et al., 2004). An early application of LT to software development was Bob Charette’s ‘lean development’ method. He advocates a two-step process that considers external influences (delivery of customer-specified value and identification of domain-specific architecture) and internal issues (effective technical development). More recently, Middleton and Sutton (2005) adopted a two-step approach to their presentation on LSD strategies. Comprehensive techniques focussing on external issues related to the elicitation of a customer’s perception of value are preceded by an in-depth treatment of internal issues addressing waste removal and effective software systems development.

The separation of concerns into external and internal goals is continued here as the initial ASD model proposed by Ågerfalk (2006) is extended to incorporate NPD and GSD challenge alleviation goals identified in chapter 3. This analysis proposes that the nine additional goals related to new product development (g1-g6) and GSD alleviation challenges (g19 - g21) may be interpreted as internally-oriented goals. Therefore, this paper does not suggest any modifications to the externally oriented goal hierarchy proposed by Ågerfalk (2006) other than to adapt the original goal codes to DScrum goal identifiers (figure 5-9).

However, modifications are suggested for internal goals. Figure 5-10 presents a proposed extension of the original internally oriented goal cluster (figure 5-9) to incorporate the expanded goal achievement hierarchy containing the additional goals. It identifies DScrum goals (table 5-2) that are deemed to have a focus on internal concerns and shows how such goals may contribute to the achievement of other DScrum goals. Dashed lines in this diagram reflect instances where an internally oriented goal either contributes to the achievement of an externally oriented goal or is supported by same. Goals may contradict other goals i.e. the achievement of one goal may be counterproductive to the achievement of another goal. Any such relationships would be indicated by a negative sign at the head of the relationship. In the internally-oriented goal rationale presented here, no such relationships are identified - hence all relationships contain a positive sign.
Figure 5-10: Goal achievement hierarchy for internally oriented DScrum goals (Adapted from Ågerfalk, 2006)

Self-reflecting teams contribute to the achievement of multi-level learning. Individual contributions to sprint retrospectives promote introspection by team members as they reflect upon insights gained during the most recent sprint. Attendance by all team members widens exposure of these insights to the entire team. The presence of external stakeholders at sprint review meetings encourages learning at the organizational level. This aspect of multi-level learning contributes to the dissemination of information to the wider organization, especially if insights reported at self-reflection meetings such as sprint reviews and sprint retrospectives result in particular development standards being established or updated. It is notable that Takeuchi and Nonaka (1986) reported on dissemination activities occurring through ‘osmosis’ (informally through judicious worker redeployment to other projects) and central standards. Certain commentators on the application of LT to manufacturing (LM) and software development (LSD) do not encourage the use of central standards (Ohno, 1988; Larman and Vodde, 2009). Although standard work practices are a valuable tenet of LT, they are established at the ‘Gemba’ or point of work by the workers and are under constant improvement through kaizen events intended to
achieve the elusive (if not impossible) LT principle of ‘perfection’. However, the supplementary Scrum practice s-11 (cf. Table 3-4) of using communities of practice to share knowledge is an information dissemination technique that is recommended within LT (Larman and Vodde, 2009). The set of goals $G = \{g_1, \ldots, g_{21}\}$ (cf. table 5-2). The goal achievement hierarchy may be expressed as a relation $\text{GAR} \subseteq G \times G =$

$$\{ (g_{18}, g_4), (g_{19}, g_{15}), (g_{19}, g_{20}), (g_{15}, g_{21}), (g_8, g_{11}), (g_6, g_{21}), (g_3, g_{11}), (g_{15}, g_{20}), (g_{15}, g_{21}), (g_5, g_{22}), (g_{11}, g_{20}), (g_{19}, g_{20}), (g_6, g_{21}), (g_{11}, g_{20}), (g_3, g_{17}), (g_{11}, g_{19}), (g_{11}, g_{21}), (g_{15}, g_{21}), (g_8, g_{21}), (g_{11}, g_{20}), (g_6, g_{21}), (g_{15}, g_{20}), (g_3, g_{17}), (g_{11}, g_{20}) \}.$$

The Cartesian product of all goals is considered in this relation as internally-oriented goals may either be supported by the achievement of externally-oriented goals $\{g_8: \text{Changes welcome}, g_1: \text{Built-in Instability}\}$ or they may support the achievement of externally-oriented goals $\{g_{11}: \text{Motivated, supported and trusted individuals}, g_9: \text{Frequent delivery of working software}\}$. The pair $(g_{15}, g_{20})$ indicates that continuous attention to technical excellence and good design supports the achievement of overcoming GSD coordination issues. An example of this would be strong functional encapsulation supported by clear interfaces that could enable effective coordination between distributed developers in their discussions on how best to coordinate the development of different components of a system. In the same way, $(g_{11}, g_{19})$ implies that achievement of the goal of ensuring that a development team consists of motivated, supported and trusted individuals will assist in overcoming GSD communication issues such as socio-cultural challenges arising from lack of trust.

The goal of exhibiting self-organization behaviour contributes to subtle control $\{(g_3, g_5), (g_{17}, g_5)\}$ as it provides opportunities for the organization to influence how a team self-organizes through effective recruitment and team member allocations. A practice that exhibits subtle control is ‘nurturing self-organizing teams’ (s-4, cf. table 3-3). Self-organization is influenced by the goal of performing overlapping development phases (g_3) as this goal results in cross-fertilization of different specialist skill sets among the team $\{(g_3, g_2), (g_3, g_{17})\}$. Not only is cross-fertilization a key characteristic of self-organizing teams, it also contributes to multi-level learning (g_5, g_4) as individuals share their knowledge. The improved flexibility of workers within
teams as a result of such cross-fertilization is a recommendation for the application of LT to LSD (Reinertsen and Shaeffer, 2005).

The internally-oriented goals that contradict the achievement of other goals may be expressed as $GCR \subseteq G \times G = \emptyset$ (as there are no internally-oriented goals whose achievement contradicts or is contradicted by other goals).

5.4 Application of framework - Value Rationale - LSD values to DScrum goals

As stated earlier, LSD values exhibit a separation of concerns in that values may be viewed as having an external or internal focus. This section reviews both sets of values separately and presents a mapping between each LSD value and any associated DScrum goals deemed to be underpinned by that value. Descriptions of both LSD values and the LSD principles used to induce them are leveraged to inform this mapping. (cf. tables 2-1 and 2-3). These LSD values are considered candidates to be present in the value base of developers.

5.4.1 Externally-oriented LDScrum value rationale

The set of externally-oriented LSD values $V_E$ identified earlier is: $\{V_1: \text{Customer Value}, V_6: \text{Product Excellence}; V_7: \text{Embrace Change}; V_8: \text{Business Environment Awareness}\}$. Let $G$ be the set of DScrum goals identified in table 5-2. The externally-oriented value rationale ($R_{vE}$) of LSD to DScrum (figure 5-11) may be specified as a relation encompassing a subset of the Cartesian product of externally-oriented LSD values and all DScrum goals: $R_{vE} \subseteq V_E \times G = \{(v_1, g_7), (v_6, g_{13}), (v_7, g_8), (v_7, g_1), (v_8, g_{10})\}$
‘Customer value’ represents the highest-order externally-oriented value guiding LSD goals and activities. This value promotes the notion that value is defined by the customer. This mandates the elicitation of not only customer requirements but also customer needs in order to ultimately provide satisfaction. Going beyond mere compliance with customer requirements is consistent with the concept of ‘big Q’ quality in order to provide customer satisfaction (Kan et al., 1994). As such, the provision of ‘Customer satisfaction’ \( (g_7) \) maps directly to this value. Customer value is also concerned with future-proofing and early delivery and efficient deployment of value. As such, it anchors the sub-value of ‘Product excellence’ which serves to address these concerns. Continuous attention to good design \( (v_6, g_{15}) \) avoids premature convergence on solutions that may hinder future maintainability needs - whether they be corrective or from a future-proofing perspective: adaptive and perfective (Lientz and Swanson, 1978). This value also promotes the delivery of functionality in change-tolerant form and as such anchors two other values: ‘Business
environment awareness’ (v₈) and ‘Embrace change’ (v₇). Awareness of potential adaptations of a product or service by a customer or market would benefit from strong domain knowledge. The sub-value of ‘Business environment awareness’ addresses this issue and is aligned with the goal of daily collaboration between business and developers (v₈, g₁₀). Change tolerance is also supported by the value ‘Embrace change’ which promotes the goals of being open to changing requirements (v₇, g₈) and maintaining controlled instability within development environment (v₇, g₁).

### 5.4.2 Internally-oriented LDScrum value rationale

The set of internally-oriented values Vᵢ presented in figure 5-12 is \{V₂: Reduce Waste, V₃: Flow of Value, V₄: Person Focus, V₅: Continuous Improvement, V₆: Data-driven Decisions, V₁₀: Effective Process, V₁₁: Effective Collaboration, V₁₂: Effective Use of Technology\}. The Value rationale of internally oriented values is obtained by mapping the internally-oriented values Vᵢ to the DScrum goals (Table 5-2). Figure 5-12 presents a mapping of internally-oriented LSD values to DScrum goals. Pursuit and achievement by a Scrum developer of these goals is motivated and influenced by their associated values. The internally-oriented value rationale is expressed as the relation \( R_{vI} \subseteq V_I \times G = \{(v₂, g_{16}), (v₃, g₉), (v₃, g_{14}), (v₄, g₃), (v₄, g₂), (v₄, g₁₇), (v₁₁, g₁₀), (v₁₁, g₂₀), (v₁₁, g₉), (v₅, g₄), (v₅, g₆), (v₅, g₂), (v₅, g₁₇), (v₁₀, g₃), (v₁₀, g₁₃), (v₁₀, g₂₁), (v₁₀, g₁₆)\} \)

The overarching internally-oriented LSD value is the removal of waste (figure 5-12). Earlier analysis of the influence of lean principles on software development indicates the need for systematic defect prevention, rework elimination and refactoring of legacy code to enhance quality and promote reusability. Such principles are addressed by pursuing a goal of ‘Simplicity’ (v₂, g₁₆) as it aims to maximize the amount of work not done i.e. remove unneeded work. Holding a value of waste reduction anchors three sub-values: ‘Flow of value’, ‘Person focus’ and ‘Continuous Improvement’.
‘Flow of value’ is an LSD value that emerged from different reports of LT in many domains. This value encourages the stabilization of development processes ensuring that only necessary demand is addressed and resources are only allocated as needed. Scope is decomposed into manageable chunks or features, thus reducing ‘batch size’. Frequent delivery of working software using sustainable development processes \{ (V_3, g_9), (V_3, g_{14}) \} will progress developers toward compliance with this value. Resource allocation is influenced by the second waste-reduction sub-value: ‘Person focus’.
‘Person focus’ promotes multi-skilling and flexibility as individuals and teams are empowered to self-organize and encouraged to display a flexible attitude toward performance of multiple roles. Individual and team empowerment is addressed by the pursuit of such goals as ‘motivated and supported individuals’ and ‘self-organizing teams’ (g_{11} and g_{2}/g_{17}). Enabling self-organizing teams may also contribute toward the goal of effecting ‘subtle control’ (g_{5}) as management may influence such self-organization through activities such as judicious recruitment or establishing provocative work targets. Self-organization and openness to work outside the confines of an individual’s core competency or role can assist in the reduction of waste. Flexibility enabled by personal attitudes to the adoption of multiple roles is influenced by the sub-value of ‘Effective Collaboration’. This value is concerned with the enhancement of communication through distance reduction and collaboration through effective role integration. As such, this value is particularly relevant to the needs of GSD and is in part addressed by overcoming GSD communication and collaboration challenges (g_{19}, g_{20}). Other goals that assist compliance with this value are ‘face-to-face work’ and overlapping development phases (g_{12} and g_{3}).

‘Continuous Improvement’ (v_{5}), relates to the active identification and resolution of issues. It also encourages the dissemination of learning to the wider organizations. Pursuing the goals of individual and team engagement in learning opportunities and dissemination of new information throughout the organization are motivated by the value of continuous improvement \{ (v_{5}, g_{4}), (v_{5}, g_{6}) \}. The use of small iterations affords opportunities for teams to reflect upon their work approaches (g_{18}). However, continuous improvement anchors another value, ‘Effective Process’ (v_{10}) which is more closely aligned with that particular goal. Another value anchored by v_{5}, ‘Data-driven decisions’ (v_{6}) addresses an LSD principle that supports continuous improvement - the pursuit of rigorous standardization in order to establish baselines for improvement. No specific DScrum goal was identified to support v_{6}. However, it was deemed reasonable to propose that v_{6} would anchor the value ‘Effective Process’ (v_{10}) as it would encourage an environment that enabled the collection and analysis of data in a systematic and scientific manner. Continuous improvement also promotes the DScrum goal of ‘built-in instability’(v_{5}, g_{1}) as the team and individual constantly seek to grow in their environment.
‘Effective process’ suggests that the method used by a team should be easily understood and employs consistent resource allocation, roles and work practices. Such needs are met by the effective application of overlapping development phases (v_{10}, g_{3}), as individuals within a team collaborate together to apply a common understanding of their development approach. Other LSD principles associated with this value are an ability to assess the team’s productivity rate and also to use fast powerful feedback loops in order to build momentum. Measuring progress through working software (v_{10}, g_{13}) and reflecting upon completed work at regular intervals (v_{10}, g_{18}) are motivated by this value. Effective process also calls for a scalable approach which is achieved within GSD by overcoming the challenges of control encountered in this context (g_{21}). It is likely that a value that promotes both scalability and establishment of productivity rates may anchor a value of ‘Effective use of technology’ (v_{11}) although no specific DScrum goals are noted for this value.

5.4.3 **LDScrum Value rationale summary**

Section 5.3 presented LSD values and DSCRUM goals in clusters that exhibit either an external or internal orientation. Both the values and goals were each presented in hierarchies in order to show how certain values may contribute to higher-order ‘broader’ values and also explain that certain lower-order goals may support the achievement of higher-order goals.

The achievement or dilution of higher-order goals is not restricted to goals within the same orientation cluster. For example, the internally-oriented goal g_{15} (Continuous attention to technical excellence and good design) supports the achievement of the externally-oriented goal g_{13} (Progress measured by working software). Conversely, the externally-oriented goal g_{8} (Welcome change) helps support achievement of the internal goal g_{1} (Built-in Instability).

This section (5.4) presented the LSD value rationale proposing that the selection of certain DScrum goals is underpinned by particular LSD values. Although certain values are defined as having an external orientation, this does not necessarily mean that such values are active in externally-oriented goals exclusively. On the contrary, V_{6} (product excellence), a value that supports the provision of customer value, is a value that may promote developers to set a target
of continuous attention to technical excellence and good design ($g_{15}$), which is an internally-oriented goal. Conversely, certain internally-oriented values promote the establishment of externally-oriented goals. The overarching internally-oriented value $V_2$ (Remove waste) encourages the establishment of the externally-oriented goal $g_{16}$ (simplicity). The internally-oriented LSD value $V_3$ (Flow of value) not only encourages the internally-oriented goals $g_{14}$ (sustainable development), it also promotes establishment of the externally-oriented goal $g_9$ (Frequent delivery of working software).

All LSD values (table 2-3) and DScrum goals (table 5-2) are represented in the value rationales presented in section 5.3. However, two of the internally-oriented values: ‘Data-driven decisions’ (v9) and ‘Effective use of technology’ (v12) do not relate directly to any specific goal. This is notable as it is an indication of an anomaly in the relationship between LSD values and distributed Scrum.

“Expressed values not operationalized into goals and corresponding method fragments are symptoms of either inadequate method descriptions or unsubstantiated claims”


However, this thesis is using the method rationale framework as a lens to explore the relationship. There is no claim that the identified LSD values should seamlessly relate to the Scrum method. Nevertheless, the above observation should be noted in any attempt to propose a method that incorporates LSD and Scrum in order to inform the method description or define the philosophy and values that anchor the method.

### 5.5 Application of framework - Goal Rationale - Method Fragments

Section 5.5 presents a method rationale for distributed Scrum. Earlier sections proposed an LSD value-anchoring hierarchy, a DScrum goal achievement hierarchy and a value rationale linking LSD values to DScrum goals (LDSScrum value rationale). This section introduces particular method fragments for distributed Scrum and proposes a method rationale for each fragment. A
method fragment may be a complete method or a component of a method. For the purposes of this investigation, three particular method fragments are identified and analysed: pre-sprint, sprint and post-sprint. Each of these method fragments is considered from a high level of granularity: the stage level. They are formed from a synthesis of the core Scrum practices identified in chapter 3 (table 3-3). It is deemed appropriate to use such a high level of granularity in this investigation as the overarching reason for the establishment of the method rationale is to explore the relationship between a set of LSD values and distributed Scrum. In-depth analysis of individual method fragments at a model, diagram or concept level would be overly specific and prove cumbersome to frame in a coherent manner in order to explore the higher-order relationships under study. These method fragments are considered from a conceptual dimension - there is no treatment of any technical aspects such as CASE tools. For the most part, these three fragments are considered from a process perspective - they are concerned with the activities carried out within these stages of the development process.

Although these three method fragments have been described above as conceptual process fragments considered at the stage level of granularity, it is important to note that other dimensions of method fragments have been considered in this research. For example, the set of core Scrum practices contains a practice called ‘sprint backlog graph’. This is incorporated into the ‘Sprint’ method fragment. However, analysis of this practice is part of the work completed to determine the DScrum goals which are ultimately linked to LSD values in the LDScrum value rationale. In method-engineering terms, using ‘sprint backlog graph’ as a method fragment would be viewed from a product perspective and at the diagram level of granularity. In some cases, it is possible that this practice might involve a tool that automatically manages calculation and communication of sprint ‘burn-down’ in which case the method fragment would be considered technical rather than conceptual. However, as stated above, in order to simplify analysis and presentation, while acknowledging that detailed analysis has considered distributed Scrum practices from different dimensions, the method fragments presented here are deemed to be conceptual, process and at the stage level of granularity. Customization of method rationale analysis to an appropriate level of granularity has been identified as an important concern for the effective use of this framework (Ågerfalk and Wistrand, 2003).
Pre-sprint practices are performed in order to achieve certain goals. Table 3-3 identifies these practices as CS-3 (create release backlog), CS-4 (create sprint backlog) and CS-5 (define sprint goal). Using these practices in core Scrum has the potential to enable a team to realize the following goals: g\textsuperscript{2}, g\textsuperscript{5}, g\textsuperscript{7}, g\textsuperscript{8}, g\textsuperscript{12}, g\textsuperscript{14} and g\textsuperscript{16}. Furthermore, if the team is using Scrum in a GSD setting, employing these practices could alleviate both communication and coordination challenges: g\textsuperscript{19} and g\textsuperscript{20}.

Using a derivation of the identifier stipulated in table 3-3, consider the three core pre-sprint practices to be listed as follows: cs\textsubscript{3} (create release backlog), cs\textsubscript{4} (create sprint backlog) and cs\textsubscript{5} (define sprint goal). GR\textsubscript{cs3} denotes the goal-rationale for ‘create release backlog’: \{(cs\textsubscript{3},g\textsuperscript{2}), (cs\textsubscript{3},g\textsuperscript{14})\}. Similarly, GR\textsubscript{cs4} = \{(cs\textsubscript{4},g\textsuperscript{2}), (cs\textsubscript{4},g\textsuperscript{5}), (cs\textsubscript{4},g\textsuperscript{7}), (cs\textsubscript{4},g\textsuperscript{8}), (cs\textsubscript{4},g\textsuperscript{12}), (cs\textsubscript{4},g\textsuperscript{14}), (cs\textsubscript{4},g\textsuperscript{19}), (cs\textsubscript{4},g\textsuperscript{20})\} and GR\textsubscript{cs5} = \{(cs\textsubscript{5},g\textsuperscript{2}), (cs\textsubscript{5},g\textsuperscript{16})\}. GR\textsubscript{SP} denotes the goal-rationale for the pre-sprint method fragment: GR\textsubscript{cs3} \cup GR\textsubscript{cs4} \cup GR\textsubscript{cs5} = GR\textsubscript{SP} = \{(cs\textsubscript{3},g\textsuperscript{2}), (cs\textsubscript{3},g\textsuperscript{14}), (cs\textsubscript{4},g\textsuperscript{2}), (cs\textsubscript{4},g\textsuperscript{5}), (cs\textsubscript{4},g\textsuperscript{7}), (cs\textsubscript{4},g\textsuperscript{8}), (cs\textsubscript{4},g\textsuperscript{12}), (cs\textsubscript{4},g\textsuperscript{14}), (cs\textsubscript{4},g\textsuperscript{19}), (cs\textsubscript{4},g\textsuperscript{20}), (cs\textsubscript{5},g\textsuperscript{2}), (cs\textsubscript{5},g\textsuperscript{16})\}.

The range of this set represents the goals that may be achieved by performance of the pre-sprint method fragment: G\textsubscript{SP} = Ran(GR\textsubscript{SP}) = \{g\textsuperscript{2}, g\textsuperscript{5}, g\textsuperscript{7}, g\textsuperscript{8}, g\textsuperscript{12}, g\textsuperscript{14}, g\textsuperscript{16}, g\textsuperscript{19}, g\textsuperscript{20}\}.

Each of the goals present in G\textsubscript{SP} may be sought by a developer based upon the values held by that developer. The potential value rationale available to developers engaged in pre-sprint activities, VR\textsubscript{SP} is as follows:

\{(g\textsubscript{2},v\textsubscript{5}), (g\textsubscript{5},v\textsubscript{3}), (g\textsubscript{7},v\textsubscript{1}), (g\textsubscript{8},v\textsubscript{7}), (g\textsubscript{12},v\textsubscript{11}), (g\textsubscript{14},v\textsubscript{3}), (g\textsubscript{16},v\textsubscript{2}), (g\textsubscript{19},v\textsubscript{11}), (g\textsubscript{20},v\textsubscript{11})\}.

The set of potential values in the pre-sprint value base, V\textsubscript{SP} is Ran(VR\textsubscript{SP}) = \{v\textsubscript{1}, v\textsubscript{2}, v\textsubscript{3}, v\textsubscript{5}, v\textsubscript{7}, v\textsubscript{11}\}.

This analysis proposes that both external and internal concerns form part of the potential value base for developers participating in pre-sprint activities. Both overarching external and internal values are identified in V\textsubscript{SP}: V\textsubscript{1} - Customer value and V\textsubscript{2} - Remove waste.
These relationships are illustrated in figure 5-13. Associations between practices and the pursuit of GSD challenge alleviation goals are identified by a dashed connector. This figure is described as a fundamental view because it illustrates the straightforward relationship proposed between pre-sprint practices and their related DScrum goals (cf. tables 3-11 and 4-10). The pre-sprint value rationale showing values that anchor each pre-sprint DScrum goal is then clearly evident. It is deemed unhelpful to incorporate additional aspects of these relationships into the model. An illustration of the complex linkages between these constructs when the goal achievement and associated value base hierarchies are integrated into the picture proves very messy. The use of appropriate diagramming techniques to support method rationale analysis is highlighted as a crucial area that needs examination. In this situation, removal of the goal achievement hierarchy is another example of framework tailoring in order to effectively use the model (Ågerfalk and Wistrand, 2003). However, because it does not add confusion to the diagram, some additional insights are provided in this model by including the value anchoring hierarchy. For the purposes of this thesis, it is deemed reasonable to employ a fundamental view as a clear starting point to relate LSD values to particular Scrum practices applied to GSD.
Figure 5-13: Pre-sprint method rationale. ‘LDScrum - Pre-sprint’ (Fundamental view)

7 Dashed line connections indicate associations with GSD goals
5.5.2 Sprint

Four core practices may be leveraged by developers engaged in a Scrum sprint: CS-6 (Lock sprint features), CS-7 (Sprint - general activities), CS-8 (Sprint - technical activities) and CS-9 (Sprint backlog graph). As stated in earlier sections, CS-7 mainly relates to the daily Scrum activity but may also cover other management tasks conducted by developers during a sprint. Technical activities is concerned with technical practices that are deemed to be particularly suited to supporting development using the Scrum framework. The reason that CS-6 and CS-9 are not subsumed into general activities is because of the specific focus placed on these practices by the founders of the Scrum framework.

Using the same derived identifier that was introduced in the previous section, each of the four sprint practices are examined: cs6 (sprint - lock features), cs7 (sprint - general), cs8 (sprint - technical) and cs9 (sprint backlog graph). GR cs6 denotes the goal-rationale for ‘lock features’: \{(cs6,g1), (cs6,g9), (cs6,g11), (cs6,g19), (cs6,g21)\}. Similarly, GR cs7 addresses the goal rationale for general sprint activities (sprint-general) = \{(cs7,g1), (cs7,g2), (cs7,g4), (cs7,g5), (cs7,g7), (cs7,g9), (cs7,g13), (cs7,g14), (cs7,g15), (cs7,g16), (cs7,g17), (cs7,g19), (cs7,g20)\}. GR cs8 presents the goal rationale of sprint-technical = \{(cs8,g2), (cs8,g3), (cs8,g4), (cs8,g5), (cs8,g9), (cs8,g11), (cs8,g12), (cs8,g13), (cs8,g14), (cs8,g15), (cs8,g16), (cs8,g17)\}. Finally, GR cs9 refers to the sprint backlog graph = \{(cs9,g2), (cs9,g14), (cs9,g19), (cs9,g20)\}. GRs denotes the goal-rationale for the sprint method fragment: GR cs6 ∪ GR cs7 ∪ GR cs8 ∪ GR cs9 = GRs =

\[
\{(cs6,g3), (cs6,g6), (cs6,g10), (cs6,g12), (cs7,g1), (cs7,g2), (cs7,g4), (cs7,g5), (cs7,g7), (cs7,g9), (cs7,g13), (cs7,g14), (cs7,g16), (cs7,g17), (cs7,g19), (cs7,g20), (cs8,g2), (cs8,g4), (cs8,g5), (cs8,g11), (cs8,g12), (cs8,g13), (cs8,g14), (cs8,g15), (cs8,g16), (cs8,g17)
\]

The range of this set represents the goals that may be achieved by performance of the sprint method fragment: Gs = Ran(GRs) =

\[
\{g1, g2, g3, g4, g5, g6, g7, g8, g9, g10, g11, g12, g13, g14, g15, g16, g17, g18, g19, g20, g21\}.
\]

Developers seek to achieve certain goals as a result of those goals being anchored by particular values held by the developer. The potential value rationale available to developers engaged in the performance of sprint activities, VRs is as follows:
\[ (g_1, v_2), (g_2, v_4), (g_3, v_5), (g_4, v_3), (g_5, v_1), (g_6, v_3), (g_7, v_4), (g_8, v_5), (g_9, v_2), (g_{10}, v_3), (g_{11}, v_4), (g_{12}, v_5), (g_{13}, v_6), (g_{14}, v_2), (g_{15}, v_4), (g_{16}, v_2), (g_{17}, v_4), (g_{18}, v_1), (g_{19}, v_1), (g_{20}, v_{10}), (g_{21}, v_{11}) \].

The range of this set represents the potential values available to the sprint value base, \( V_S = \text{Ran} (\text{VR}_S) = \{ v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_{10}, v_{11} \} \).

In keeping with the analysis of the previous method fragment, both external and internal concerns form part of the potential value base for developers participating in sprint activities. Both ‘V_1 - Customer value’ and ‘V_2 - Remove waste’ are proposed in \( V_S \). These relationships are illustrated in figure 5-14. Practices motivated by GSD challenge alleviation goals are identified by dashed-line connections. The model is considered a fundamental view of the sprint method fragment - this figure simply presents a direct relationship between core sprint practices and the goals that they pursue. It does not take into account the goal achievement hierarchy that may lead to higher-order goals being achieved as a result of additional values from the overall LDScrum value base being present in the developer. It is a direct mapping of each method fragment to potential goals that may be achieved if particular values are present to anchor the desire to achieve those specific goals. In this case, the relevant goals are \( g_1, g_2, g_3, g_4, g_5, g_7, g_9, g_{11}, g_{12}, g_{13}, g_{14}, g_{15}, g_{16} \) and \( g_{17} \). Furthermore, in a GSD setting, these practices may help to achieve all three GSD goals: \( g_{19}, g_{20} \) and \( g_{21} \). The strong coverage of potential goals to be achieved serves to reinforce the notion of the sprint being at the heart of the Scrum development method.
Figure 5-14: Sprint method rationale. ‘LDScrum - Sprint’ (Fundamental view⁸)

⁸ Dashed line connections indicate associations with GSD goals
5.5.3  **Post-Sprint**

Two core practices may be leveraged by developers engaged in the post-sprint method fragment: CS-10 (Review) and CS-11 (Retrospective). Using the same derived identifier that was introduced in previous sections, each of these practices is examined. GR\textsubscript{cs10} denotes the goal-rationale for ‘review’: \{(cs\textsubscript{10},g\textsubscript{1}), (cs\textsubscript{10},g\textsubscript{6}), (cs\textsubscript{10},g\textsubscript{7}), (cs\textsubscript{10},g\textsubscript{10}), (cs\textsubscript{10},g\textsubscript{12}), (cs\textsubscript{10},g\textsubscript{13}), (cs\textsubscript{10},g\textsubscript{19}), (cs\textsubscript{10},g\textsubscript{20}), (cs\textsubscript{10},g\textsubscript{21})\}. Similarly, GR\textsubscript{cs11} addresses the goal rationale for retrospective: \{(cs\textsubscript{11},g\textsubscript{2}), (cs\textsubscript{11},g\textsubscript{4}), (cs\textsubscript{11},g\textsubscript{6}), (cs\textsubscript{11},g\textsubscript{12}), (cs\textsubscript{11},g\textsubscript{18}) (cs\textsubscript{11},g\textsubscript{19}), (cs\textsubscript{11},g\textsubscript{20}), (cs\textsubscript{11},g\textsubscript{21})\}. GR\textsubscript{PS} denotes the goal-rationale for the post-sprint method fragment: GR\textsubscript{cs10} \cup GR\textsubscript{cs11} = GR\textsubscript{PS} = \{(cs\textsubscript{10},g\textsubscript{1}), (cs\textsubscript{10},g\textsubscript{6}), (cs\textsubscript{10},g\textsubscript{7}), (cs\textsubscript{10},g\textsubscript{8}), (cs\textsubscript{10},g\textsubscript{10}), (cs\textsubscript{10},g\textsubscript{12}), (cs\textsubscript{10},g\textsubscript{13}), (cs\textsubscript{11},g\textsubscript{2}), (cs\textsubscript{11},g\textsubscript{4}), (cs\textsubscript{11},g\textsubscript{6}), (cs\textsubscript{11},g\textsubscript{12}), (cs\textsubscript{11},g\textsubscript{18}), (cs\textsubscript{11},g\textsubscript{19}), (cs\textsubscript{11},g\textsubscript{20}), (cs\textsubscript{11},g\textsubscript{21})\}.

The range of this set represents the goals that may be achieved by performance of the post-sprint method fragment: G\textsubscript{PS} = Ran(GR\textsubscript{PS}) = \{g\textsubscript{1}, g\textsubscript{2}, g\textsubscript{4}, g\textsubscript{6}, g\textsubscript{7}, g\textsubscript{8}, g\textsubscript{10}, g\textsubscript{12}, g\textsubscript{13}, g\textsubscript{18}, g\textsubscript{19}, g\textsubscript{20}, g\textsubscript{21}\}.

Each of the goals present in G\textsubscript{PS} is pursued by a developer as a result of being anchored by particular values present in the value base of that developer. The potential value rationale available to developers engaged in the performance of post-sprint activities, VR\textsubscript{PS} is as follows:

\{(g\textsubscript{1},v\textsubscript{7}), (g\textsubscript{2},v\textsubscript{4}), (g\textsubscript{2},v\textsubscript{5}), (g\textsubscript{4},v\textsubscript{5}), (g\textsubscript{5},v\textsubscript{1}), (g\textsubscript{6},v\textsubscript{7}), (g\textsubscript{10},v\textsubscript{8}), (g\textsubscript{11},v\textsubscript{11}), (g\textsubscript{12},v\textsubscript{10}), (g\textsubscript{13},v\textsubscript{10}), (g\textsubscript{14},v\textsubscript{11}), (g\textsubscript{15},v\textsubscript{11}), (g\textsubscript{20},v\textsubscript{11})\}.

The range of this set represents the potential values available to the post sprint value base, V\textsubscript{PS} = Ran (VR\textsubscript{PS}) = \{v\textsubscript{4}, v\textsubscript{5}, v\textsubscript{7}, v\textsubscript{8}, v\textsubscript{10}, v\textsubscript{11}\}.

This analysis proposes that both external and internal concerns form part of the potential value base for developers participating in post-sprint activities. However, the fundamental mapping only shows a direct mapping in V\textsubscript{PS} to the overarching external value: V\textsubscript{1} - Customer value. Although not explicitly mapped, the overarching internal value, V\textsubscript{2} - Remove waste, is supported by the mapping of post-sprint goals to certain sub-values in the internal LSD value base hierarchy (cf. figure 5-8): V\textsubscript{4}, V\textsubscript{5}, V\textsubscript{10} and V\textsubscript{11}. These relationships are illustrated in figure 5-15.
Figure 5-15: Post-sprint method rationale. ‘LDScrum - Post-sprint’ (Fundamental view)
5.6 Summary

Chapter 5 proposed a framework to be used as a lens for empirical research into the relationship between LSD and the application of Scrum in GSD. The concept of methods and method tailoring was described. Method engineering concepts were explained at a high-level in order to introduce certain concepts used to assemble methods from components called ‘method fragments’. A specific dimension in method analysis was examined: method rationale. The method rationale analysis framework was described. Components of this framework include values, goals and method fragments. LSD values and distributed Scrum goals were analysed in order to relate them effectively for use within the framework. The core Scrum goals introduced in earlier chapters were synthesized into three method fragments: pre-sprint, sprint and post-sprint. A method rationale was proposed for each of these method fragments, relating them to particular goals and by association, certain LSD values. These proposed method rationales make up the candidate LDScrum model which forms the conceptual framework used to support empirical research. This research design and execution is presented in the upcoming chapters.
Chapter 6- Research Strategy

6.1 Introduction

Figure 6-1 highlights the work reported in chapters 6 and 7 in the context of the overall work breakdown structure (Figure 1-7). This chapter addresses strategic research considerations (work package 2.1.1). Although research considerations influenced the work conducted to build the conceptual model, it was deemed more effective to present all research design work under the empirical research WBS segment. The primary purpose of a WBS is to determine all project scope. As such, dependencies are not a key issue when using this structure (PMI, 2008).

Figure 6-1: Research approach WBS segment (research philosophy work package)

A critical aspect of any research endeavour is that the research process is clear and well understood by the researcher. This goal may be achieved by pursuing the following three steps (Cua and Garrett, 2008; Shanks, 2007):
(i) Leverage a framework that enables the researcher to identify the nature of the reality under research (the ontology).

(ii) Examine and appreciate the relationship between the researcher and the knowledge to be acquired (the epistemology).

(iii) Determine a clear approach on how to uncover that knowledge (the methodology).

This researcher has adopted an interpretive stance to conduct this study. This approach is adopted because of his belief that the pursuit of scientific knowledge on ISD relies on the interpretations and perceptions of that community (or social system) and is best served by rich descriptions of that environment (or reality). The main purpose of this chapter is to present this researcher’s understanding of the ontological, epistemological and methodological landscape and explain the reasoning process that led to his interpretive epistemological preference.

In order to expound on the researcher’s value-laden interpretive preference, this chapter is introduced by explaining the motivations and important activities that informed the objectives of this scientific inquiry. As stated above, a vital aspect of this work is a well-understood research process. A basis for the establishment of a clear understanding of a single research process is to gain a broad grasp of the wider scientific research field. To this end, section 6.2 presents a broad overview of scientific research, placing special emphasis on the field of information systems research in deference to the area of inquiry covered by this thesis. This includes an overview of both ontology and epistemology in order to support an explanation of the researcher’s philosophical assumptions with regard to these two concepts. Section 6.3 describes epistemology in greater detail in order to flesh out the issues of appropriate epistemological approaches, especially in relation to IS research. This section presents a detailed account of the rationale used to determine the researcher’s epistemological stance. Finally, section 6.4 presents a description of different methodological options followed by a rationale for the selection of the case study method to perform this research. In order to summarize all of these topics and justify the selected research approach, this section concludes by using an interpretive research evaluation framework to describe the overall research design.
6.1.1 Research Objectives

At the outset of this research process, an appreciation for ASD honed from many years of industrial software development was augmented by participation in a European ASD research project (FLEXI\(^9\)). Insights gained from this research included awareness of ASD research challenges and the identification of an important ASD-related topic: LSD. The researcher has much experience providing lean solutions to the global manufacturing community through the development of software products that incorporated features such as Kanban and Just-in-time (JIT). As a result, the identification of a research need to resolve the confusion between LSD and ASD was extremely interesting and relevant to this researcher. This early work led to the establishment of a high-level core research objective - exploration of the relationship between LSD and ASD.

In order to narrow the focus of such a wide brief, it was deemed prudent to concentrate on specific aspects of ASD. Having spent over fifteen years working in GSD, the researcher recently completed an MSc dissertation that enabled him to formalize theoretically many of these experiences. This background coupled with the identification of GSD as an area that requires further research within ASD (Abrahamsson et al., 2009) motivated the convergence of the ASD aspect of this research into the use of Scrum in GSD. (Scrum was selected due to its dominance in ASD (Dyba and Dingsoyr, 2008) and to narrow the ASD focus). Therefore, this study examines how SCRUM may leverage the values of LSD when used to manage GSD. This domain may be described as a social system as it involves interaction amongst various actors. Given that the research subject examines ISD, this study is categorised under the heading 'IS research'.

\(^9\) FLEXI was part of the ITEA 2 (Information Technology for European Advancement) project. Its remit was the exploration of Agile methods with a special focus on scaling agile (or "Agile in the large").
6.2 Science and scientific research

In this section, the broad subject of scientific research is described and distinctions between studies of nature and social systems are discussed. This is followed by an overview of the subject of ontology. Having discussed a number of different philosophical viewpoints in relation to the nature of reality, the ontological assumptions underlying this research study are declared. A similar approach is taken to the area of epistemology. The section concludes by introducing two opposing epistemological stances that have caused much debate in the IS research community.

6.2.1 Natural v Social Science

Scientific research sets out to create, support or reveal valid knowledge. The validity of knowledge is dependent upon acceptance by the validating group of the values that underpinned the approach taken to generate such knowledge. Such value sets are referred to as research paradigms. Appropriate values have been the source of much discourse through the centuries. In their review of IS research, Kanellis and Papadopoulos (2008) explained that three of the key questions that have dominated this area are:

- What is to be studied - what is the nature of existence?
- Considerations about who is performing the examination.
- How knowledge is acquired.

The first of these points relates to ontology - the nature of existence. The second and third issues refer to epistemology - the position taken about the appropriate approach to adopt in order to produce valid knowledge. In relation to the ontological questions, much debate on this subject has centred on whether reality consists of objective physical entities and structures that support such entities or whether reality also comprises of a subjective intellectual or spiritual component and as such is tied to interpretation by different observers. In relation to the epistemological points, the former notion of reality lends itself more cleanly to the positioning of the observer or researcher as being independent from the research process. It also seems to suit the application of a consistent logical analysis of reality using causal relationships to generate laws in a
deterministic fashion that constitute knowledge of that reality (positivism). A reality of objective physical entities appears to fit neatly into examinations of the world of nature (natural science).

A somewhat messier picture emerges when the research seeks to produce knowledge about humans and human interactions within social systems (social science). The latter notion of reality as containing a subjective interpretive judgemental element appears to sit better with investigations into the individualistic complex entities that are humans interacting within social systems. It may be impossible for such investigations to truly be independent from the researcher or indeed it may be preferable that the researcher’s judgement is applied in order to enrich analysis of the data being examined. The complexity of a reality that consists of multiple interpretations of multiple perceptions of entities within a group setting may not lend itself to the restrictions of deductive reasoning based upon causal relationships (Kanellis and Papadopoulos, 2008).

The IS domain is very diverse and presents both technical and social dimensions. Such diversity may contribute to the reason why no specific IS research paradigm has emerged. Analyses of journal publications in IS research have revealed early dominance of positivism although other approaches have become more prevalent in recent years. (Kanellis and Papadopoulos, 2008).

The set of specialist IS sub-communities that reflect the diversity of this discipline lack a common understanding or ‘conceptual platform’ in which to unify and build their research output. Many commentators propose that such a common conceptual platform is not possible and contend that pluralism of ideas, viewpoints and research approaches is a positive element within a research community. However, pluralism relies on a core foundation of knowledge that a community may leverage to identify themselves as members of an overarching common area of interest (or research). The absence of this foundational reference results in the IS research community being fragmented and in need of effective internal and external communication channels. It is likely that such fragmentation is not a positive contribution to ongoing debates and disagreements around effective epistemological stances within IS research and is part of the reason that an IS research paradigm does not appear to be evolving (Hirschheim and Klein,
Ontology, epistemology and their application within IS research are elaborated in sections 6.2.2 and 6.2.3.

6.2.2 The Ontological Question

A sound epistemological foundation to potential methodologies can inform the researcher’s selection of a method based upon their own assumptions of the particular IS phenomenon they are studying and how they believe they should elicit knowledge from that domain. Essentially, the researcher’s understanding and assumptions about ontology (or the reality that they are studying) influences their epistemological stance. This section explains the researcher’s understanding of general philosophical ontological assumptions so that he may form and declare his own philosophical ontological assumptions in relation to the reality or aspect of the IS domain that is being researched.

Ontology is concerned with the philosophy of existence. It addresses the nature of reality. Philosophers have put forward various theories addressing ontological issues. These theories may be grouped into three categories: ‘idealism’, ‘realism’ and ‘phenomenology’. Idealism stems from the word ‘idea’ and contends that reality is a mental construct and that there is no ‘objective reality’. It may be further sub-divided into theories around subjective and objective idealism. Subjective idealism proposes that reality consists only of objects that may be perceived and that all reality is a reflection of ideas stemming from experiences. Objective idealism, while still proposing that reality is a mental construct, questions the notion that it is an idea that has been experienced - it submits that there is an objective reality which is based upon ideal concepts that may or may not be perceived by individuals. (Such an objective reality may be formed if one were to consider an omniscient presence that perceives all things in their ideal state - hence placing them in reality). In direct opposition to idealism, realism is based on an objective reality that exists outside of the mind and exists separately from people’s experiences. There are different nuances to realism also. The theory of phenomenology bridges both idealism and realism - it accepts that reality is composed of two entities: noumena and phenomena. The former are entities that exist but are unknown and the latter are entities that have been leveraged by the mind in order to create knowledge about that entity. Phenomenology addresses the
acquisition of knowledge and as such does not emphasize the nature of reality - it is more concerned with the relationship between the knowledge-seeker and the reality being learnt. It describes ‘phenomenological reduction’ as the process whereby a person may reflect upon their experiences in order to make judgements based upon their own values and beliefs. This establishment of meaning associated with an experience is termed ‘intentionality’ (Kanellis and Papadopoulos, 2008).

Mingers (2001) interpreted the work of Habermas’ theory of communicative action to summarize the ontological dispersion that must be addressed by research. There are multiple realities that exist and that require different treatment in order to comprehensively examine a situation. The material world is independent of humans and is subject to observations (realism). Human capabilities for self-reflection have given rise to the personal world where individual experience, background and beliefs inform the reality (idealism). There exists an inter-subjective or social world that is constantly forming and re-forming as a result of interactions between participants in that social reality.

Figure 6-2: Summary of the three worlds (Mingers, 2001)
In stating their position on ontology, Kanellis and Papadopoulos (2008) echoed the views of Mingers (2001) by taking a phenomenological stance that there is a physical objective ‘external’ reality that is perceived by individuals and is then formed or ‘interpreted’ into a concept in the mind as a result of their own background and experiences (a type of external-internal transformation process). They posited that this physical objective reality is constant, leading to most members of a social community sharing a common perception of the physical entity - it is their interpretation that creates the individual subjective reality. This is contrasted with social reality where the entities being perceived are not physical and perceptions are constantly changing due to interactions among the social community. Thus, there is additional complexity in social reality as entities within this reality are continuously formed and re-formed thus diluting the potential for shared perceptions. The reality is socially constructed. IS is described as a large social system where social reality is both created and observed by humans.

This researcher also adopts a phenomenological view of reality. He believes that noumena and phenomena both exist. Perceptions of entities lead to internal analysis and a subjective reality is conceived by each perceiver based upon their own background and values. A key motivation for this assumption is because a belief of the existence of an objective reality outside of the mind (realism) is not comprehensive enough to support this research. Within the context of many individual members of a community, there may be a perceived objective reality that exists outside the mind of a member, but the concept of a social reality relies on some other member perceiving that reality within their mind. The reality being examined is an ISD organization that is engaged in collaborative activities in order to produce a software product. This is an inquiry into humans and human social communities (or ISD teams), which constitutes a social reality and as such is primarily concerned with team member’s perceptions and interpretations of their environment. The phenomenological concept of intentionality is very suited to such research as it accepts that individual’s give meaning to their reality. This research seeks to interpret the meanings applied by individuals within a social reality when performing certain activities in development situations.
6.2.3 **The Epistemological Question**

Epistemology is the study of how human knowledge is constructed. At its core, it attempts to determine whether or not it is possible to acquire knowledge about reality. Dogmatists claim that the possibility of such knowledge acquisition is true and does not need to be questioned. In contrast, scepticism questions the premise that knowledge about standalone entities of reality is attainable given that we perceive these entities in relation to their context. Pragmatists concentrate on whether the knowledge gained is useful and improves our lives (Kanellis and Papadopoulos, 2008). Hirschheim (1992) identified the definition of knowledge as one of two main concerns within epistemology. In describing knowledge he explained that the Greeks believed that science was a process that transformed ‘doxa’ (that which was believed to be true) into ‘episteme’ (that which was known to be true). The major problem with this process is establishing that which is known to be true since individual judgement is restricted by language, culture and experiences. An acceptance emerged from this dilemma that knowledge is asserted and is not absolute - it is conditional on societal or group acceptance. Knowledge assertions are supported by evidence and claims are established based upon various agreed conventions. This means that knowledge may change over time based upon the emergence of new counter-claims related to the accepted knowledge.

Part of the discussion on knowledge acquisition centres around potential methods that may be used to achieve this goal. A key concern in this regard considers whether intellectual capabilities are the primary source of knowledge and are present in the individual before or despite experience (a priori) or whether experience is the key source of knowledge and may use intellect to process the experience (a posteriori). In this respect, there is much overlap in the discourse on various methods. Rationalism purports that the intellect operates according to its own capabilities and does not rely upon experiences or at most is prompted by experiences to apply pre-wired (a priori) intellectual abilities. This would relate to knowledge generation via dialectics where the application of reason results in acceptance of a truth. Empiricism proposes a directly opposing view that places experiences to the forefront of the search for truth by stating that reality can only be learnt through experiences and that the intellect’s role is to process information provided
through the senses. In essence, the debate is about the dominance of reason over experience or vice-versa (Kanellis and Papadopoulos, 2008).

Rather than adopting an extreme stance on either side of this debate, Kanellis and Papadopoulos (2008) took the epistemological position that there is a need to involve both experience and reason when developing knowledge. They emphasised that when dealing with social reality, (as exists in IS), the intellect must analyse and process the diverse set of perceptions that arise within this rapidly changing reality, in order to formulate a conception that will contribute to the knowledge of that reality. This researcher would suggest that this concept should be taken further as the intellect of a ‘central processor’ or researcher is required to structure and reason out meanings from the perceptions and conceptions produced by different members of the social community. A major question in relation to this central processor (or researcher) is how their intellect would rationalize the commonality in order to generate knowledge of the social reality. Would it reflect positivist thinking and be a mathematical logical reduction of member’s conceptions to deduce a universal predictable law about the social reality? Or would it be an interpretive analysis of conceptions that relies upon the experiences, capabilities, values and morals of the researcher in order to reveal some insight about the social reality that results in knowledge that may help grow understanding of this situation? The position adopted by this researcher is the latter. However, this particular decision relates directly to the second key epistemological concern identified by Hirschheim (1992): the manner in which ‘valid knowledge’ is obtained. This concern is addressed by the application of science and science may be defined as the ‘search for understanding’. This broad definition encompasses both normal and pseudo-science i.e. universally accepted rigorous replicable methods of inquiry as against open-ended examinations of a subject through processes such as meditation. In the West, the definition of science has been more restrictive resulting in the positivistic scientific method being considered the acceptable convention for scientific pursuits. This method posits beliefs that are empirically tested.

This overview of epistemology opened with the views of the early Greek philosophy movements. In his historical review of the quest for valid knowledge, Hirschheim (1992) also referred to
these groups. He identified a dichotomy of two forces in early Greek philosophy: ‘believers’ (or dogmatists) and ‘doubters’ (or sceptics). Dogmatists based their thinking on Plato and Aristotle and were responsible for the early development of positivist thinking. Sceptics can be traced back to the sophists who posed counter-arguments and had an early influence on the consideration of alternative ways to acquire knowledge. Positivism has proven very successful in its application in the physical sciences but has caused controversy when used in the social sciences. This issue is explored further in the next section as it has emerged as a recurring theme within epistemology, especially in social science.

6.3 Rationale for Epistemological Preference

This section explores the issue of knowledge acquisition through either rigorous deductive approaches or interpretive inductive mechanisms. General epistemological concerns are considered in the context of IS research and also in relation to this particular ISD investigation. The section begins by examining the epistemology of positivism. The evolution of this dominant scientific approach is outlined and weaknesses identified with its application to social science (and IS research) are highlighted. Section 6.3.2 reviews the opposing stance of interpretivism. This is followed by a discussion on which of these two epistemological preferences is most appropriate to IS research. This section concludes by presenting the researcher’s epistemological assumptions based on the general points expressed in the previous sections.

A variety of epistemological stances have been adopted within IS research over the past forty years. Early publications showed evidence of two research approaches: empirical research that applied the scientific method and descriptive accounts of different IS topics. Examinations of major IS journals have shown that research paradigms designed to meet the needs of IS research have not emerged and that the majority of research designs are positivist in nature. A number of major themes are evident in IS topics and one of these, ‘engineering and design’ represents the area being examined in this thesis. A key requirement for IS research is that the researcher adopt
and explain their epistemological position. The core dichotomy to be addressed is whether their approach is positivist or anti-positivist (Kanellis and Papadopoulos, 2008).

### 6.3.1 Positivism

Positivism is associated with a particular ontological stance: ‘realism’. As discussed earlier, this postulates that reality consists of objects that are independent of any subjective interpretation by individuals (including the researcher who is seeking knowledge on a particular domain or topic).

Positivist methods seek causal relationships between variables. They are characterised as rational approaches that require logical thinking to disprove hypotheses. The researcher does not get directly involved in the research process. He/she may construct theoretical models that facilitate measurement of variables in a quantifiable manner. Inferences may be drawn from the sample studied to the general population or domain under research. Hirschheim (1992) summarized various contributions on this subject by characterizing the basis for positivism under five headings:

(i) **Unity of the scientific method**: The scientific method is a valid approach for acquiring all knowledge, regardless of the objects being examined.

(ii) **Search for causal relationships**: Using reductionism, a domain or set of objects under examination may be decomposed into their constituent parts in order to find causality between relationships and establish the presence of consistency within the domain.

(iii) **Belief in empiricism**: Valid knowledge is obtained from the senses - subjective perceptions and unconscious intellect are not deemed valid.

(iv) **Science is value-free**: The researcher is totally independent from the research. The subjective values and beliefs of the researcher do not influence the process undertaken to acquire valid knowledge.

(v) **Science is based upon logic and mathematics**: Mathematics provides a formal notation that enables quantitative analysis to support the search for causal relationships.
In order to structure his treatment of the evolution of epistemology, Hirschheim (1992) presented five eras of epistemological thinking that relate to positivism: Positivism; Anti-positivism; Neo-positivism; Contemporary critics and Post-positivism. A single theme that runs through the major issues addressed by these different eras is the struggle to apply logical rigorous deductive concepts to the exploration of social sciences. Positivism emerged as a rigorous scientific approach during the renaissance period and established a separation of concerns in relation to the examination of reality. Study of the mind and soul should be the remit of theology and the study of matter be addressed by science. This established the notion that objects of nature were physical entities that are distinct from concepts of the mind such as consciousness or awareness and promoted scientific research of nature to be independent from the researcher. The application of positivism to inquiries other than physical science led to the emergence of social science - the study of human behaviour. Scientists proposed the application of causation and empiricism to social sciences and promoted the view that the five senses were the only consistent and pure way to perceive data and elicit knowledge. The anti-positivism movement emerged to counter the lack of regard for meaningful experience by positivists in their inquiries into human systems. A key tenet of anti-positivism is the augmentation of sensory perceptions (experience) with an appreciation or examination of the meaning, understanding or interpretation of these experiences. Contributors to the anti-positivism movement propose that two different categories of knowledge should be sought depending upon the research subjects: explanatory knowledge and understanding knowledge. Explanatory knowledge is described by ‘nomoethic’ laws that reflect physical causation and fit the consistency of physical science whereas the pursuit of understanding relies on deep meanings and ‘idiographic’ research approaches should be applied when addressing human science. The requirements of this investigation reflect such a need for understanding in order to detect LSD influences in the team’s application of Scrum.

A refined form of positivism, neo-positivism has become the dominant epistemology of modern science (Hirschheim, 1992). Phenomenalism - the restriction that reality emanates from phenomena that are experienced was widened to embrace physicalism where objects of reality are present in the world even though they may not have been experienced. Intersubjective agreements on objects is considered acceptable. Another notable refinement was the
modification of the scientific method from individual explanation to theoretical knowledge statements linked by deductive logic to be falsified by direct observation. This model is seen today in the hypothetico-deductive model used as the scientific method in the majority of research currently performed. Much criticism of neo-positivism emerged in relation to its applicability within the social sciences. The notion of value-free inquiries is questioned as arguments have been made that researchers should use their values as part of their research into social systems. The application of values may enrich the theories to be examined or discarded. Conversely, social research may be diluted or misguided by a value-free analysis of a system as such analysis may fail to uncover or address underlying influential values within the system. Another contemporary criticism of neo-positivism is the unity of the scientific method. This claim that all science may be predicted and generalized via a cycle of observations, hypothesis, testing and analysis is refuted in relation to social sciences. Human actions involve social meaning that requires richer deeper examination than is possible from observations. Social science needs to inspect more than just physical aspects of its subjects - it requires an understanding of the mental processes that influence social activity.

The post-positivism era has resulted in much discussion about the need for a different approach to the accepted scientific method of generating or augmenting knowledge. This debate has brought into question the fundamental notion of building knowledge from demonstrable propositions (apodictic knowledge) and asserts that knowledge claims should arise from acceptance by the community (possibly resulting from a reasoning process, which would constitute dialectic knowledge). This runs contrary to the positivistic goal of ‘completing’ knowledge about some reality and acknowledges that knowledge has a flexibility that may result in it being true although it cannot be proven true at the time that the research is conducted. Post-positivism disagrees with the pursuit of a ‘single true paradigm’ to address the generation of knowledge. It proposes a methodological pluralism to address the difficulties that arise from applying a method that is effective in one research situation but problematic in another scenario, e.g. the positivistic scientific method may hold up well when applied to a natural physical phenomenon such as a plant, whereas the application of this approach to human or social
research is challenged by the need to incorporate mental, individual subjective components into the research process (Hirschheim, 1992).

Kanellis and Papadopoulos (2008) summarised the views of various IS researchers on the fragility of reliance on causality and determinacy to establish knowledge in this field. Organizations are subject to continual change and the linkage between cause and effect can be extremely complex thus negating theories that have been verified using invalid causality. The reduction of organizations into independent parts whose behaviour can be theoretically analysed, predicted and proven is subject to pressure from the fast-moving environments in which they reside. These views align with post-positivism in that they propose that research in modern social systems should adopt a divergent and more open attitude to knowledge acquisition rather than a convergent deductive closed outlook.

This research investigation sets out to gain an understanding of the reasons that various individuals within an ISD team perform particular activities. As such, this could be termed an endeavour in social science. The description above of the five epistemological eras reported a consistent vulnerability in regards to the application of positivism to social science. Furthermore, the activities under investigation are being executed within a fast-moving volatile environment that is subject to both internal and market pressures. As stated above, it is preferable to adopt an open attitude to knowledge acquisition in this context.

6.3.2 **Interpretivism**

Given that the motivation for post-positivism appears to be the inability of positivism to meet all research situations, it would appear useful to reflect on the anti-positivist movement when considering methodological pluralism. That movement was concerned with the need to acquire understanding and apply meaning to the reality being studied. This reveals a need to interpret the reality and leads to an interpretivist epistemological stance. Interpretivism believes that knowledge is gained by developing a deep and rich understanding about a subject area - not by the theoretical modelling of causal relationships. To support this concept, the researcher is part of the research process and uses their judgement to analyse data uncovered. Interpretivism in IS
research can be viewed as ‘value-rich’ (as against the value-free stance of positivism). It relies upon an assumption that reality is formed by humans interacting within social groups. The creation of knowledge resulting from an enquiry into this reality leverages both the values of the researcher and those of the humans being studied. The researcher’s values inform him on how he conducts his work. Values, beliefs and culture of the research subjects determine how they construct the social reality upon which the research depends (Walsham, 1995a).

As part of their proposal on the generalization of research, Lee and Baskerville (2003) considered the views of various social scientists (Schutz, Geertz, Van Maanen) in relation to interpretive research. Data collected from research subjects reflects their understanding of their reality - it constitutes a first-order construct and contributes to a rich description of that world. It is influenced by the values of the research subjects (although the values of the researcher are also at play here as he/she determines what data collection approach to select, apply and modify during the data collection process). It should be noted that this rich description is still restricted to being a description of the meanings applied by the subjects based upon their experiences, backgrounds and situations at that time. The analysis of first-order constructs in order to elicit knowledge from the combined realities (and thus present a general description of the social reality) results in a second-order construct that is subject to the understanding and values of the researcher. The reasoning applied by the researcher to generate the more general second-order construct is the result of ‘analytic generalization’ (Yin, 2009). It forms a general theory rather than just a thick description and it may inform future interpretive researchers on the phenomenon under research (Lee and Baskerville, 2003).

6.3.3  **Appropriate stance to IS Research: positivism or interpretivism?**

The previous two sections have described two opposing epistemological stances: positivism (or prediction/reduction) and interpretivism (or interpretation). The opposing positions of ‘hard’ predictable positivism and ‘soft’ messy interpretivism have clearly caused much discourse and difficulty among the IS research community. It has been described as the ‘rigour versus relevance’ debate i.e. the restrictions of positivistic determination can make the research question too shallow whereas interpretation that does not rely upon some analysis process could result in
an unmanageable anarchy of knowledge claims (Fitzgerald and Howcroft, 1998). This section opens by accepting the initial dominance of positivism and explores the acceptance of interpretivism within the IS community. It then discusses different applications of these two epistemological preferences to IS research. Finally, a framework is presented that positions each of the epistemological stances and their associated methods in order to support a pluralistic approach to the multi-disciplinary nature of ISD organizations.

As stated earlier, scientific knowledge is based upon acceptance by the scientific community on the approaches taken to transform assertions or claims into scientific facts. A major challenge in the context of IS research is the epistemological controversy regarding positivism/interpretivism. Walsham (1995a) used insights from journal editorial policies regarding epistemological stances to examine this controversy. He noted that although positivism has dominated these journals, interpretivist approaches are becoming more prevalent. Interpretivism is an acceptable approach within influential journals and enjoys support among researchers in diverse IS sub-topics including systems design, artificial intelligence and IS management. A comparison of the editorial policy of one of the key mainstream IS journals (MIS Quarterly) revealed a clear change from strict positivist guidelines in 1989 to a much wider approach that embraced both positivist and interpretivist approaches three years later. Walsham (2006) provided more recent evidence of this tendency in his report on IS journal publications of interpretive studies. Another indication of the increasing acceptability of interpretivism is the emergence of researchers who consider a more unified approach to research. In contrast with the positivist diktat of the unity of the scientific method, the unified approach proposes that interpretivist and positivist approaches should be combined to enable a rigorous examination of hypotheses generated from a rich understanding of a phenomenon (Walsham, 1995a).

Four strategies are suggested that incorporate the various approaches proposed on how to address the positivism/interpretivism debate within the IS community.: “supremacism”, “isolationism”, “integrationism” and “pluralism” (Fitzgerald and Howcroft, 1998, p. 320).
(i) **Supremacism** seeks the establishment of one overarching paradigm for IS research. A concern with this view is the potential restriction of innovative designs.

(ii) **Isolationism** promotes a dogmatic viewpoint that there is only ‘one true way’. This strategy would achieve paradigmatic closure but at the cost of ignoring strengths observed with one or other of the stances.

(iii) **Integrationism** recommends an analysis approach that combines both interpretivism and positivism. The opposing character of each paradigm would cause difficulties on how best to establish the suitability of a particular integrated approach.

(iv) **Pluralism** refers to studies that leveraged both ‘hard’ and ‘soft’ approaches. Different views exist on how pluralism should be applied. One view proposes the complementary approach of using interpretivism for exploration and positivism for subsequent theory confirmation. Another view accepts the ontological and epistemological distinctions between the paradigms but suggests that such distinctions should not be completely maintained when applied at the methodological level. The application of multiple methods to cover different dimensions of the area under research is further supported by (Mingers, 2001).

Fitzgerald and Howcroft (1998) reflected upon the weaknesses of the different strategies and concluded that the resolution to this debate may only be realized by the adoption of a more ecumenical spirit. This spirit should not try to justify hard or soft approaches but embrace the challenges of the diverse IS research agenda and consider research approaches from that context rather than pre-defined scientific methods. Of the four categories presented above, the pluralist viewpoint appears to be most closely aligned with this aspiration. Although an interpretive epistemological stance guides this research study, there are elements of the detailed design that relate to a “pluralist ecumenical accommodation at the lower methodological level” (Fitzgerald and Howcroft, 1998, p. 322). The aforementioned second-order construct of the social reality studied is formed using techniques borrowed from the positivistic hypothetico-deductive model. It is intended that the transparency and consistency afforded by the use of these techniques supports the ‘analytic generality’ of this work (Lee and Baskerville, 2003) while also addressing the ‘relevance v rigour’ concern.
Although the discussion to this point has focused on the positivism/interpretivism debate, a third stance that has emerged is ‘critical theory’. Braa and Vidgen (1999) posited that the inclination or requirement that researchers embrace one of these three particular epistemological positions (and their associated research methods) is restrictive to the examination of the multi-disciplinary nature of IS in organizations. Their belief that the optimal IS research laboratory is the organization motivated them to establish a research framework that would consider the various methods that could be integrated in order to overcome such restrictions and incorporate the benefits realized by the application of interpretivism and positivism to a situation as appropriate. This is consistent with the pluralist strategy reported above. They proposed that a key outcome of positivistic research is predictability and repeatability. In order to achieve this, the positivistic researcher is external to the context and aims to control the environment, typically engaging in the act of *reduction* in order to ensure reliable predictions are generated. On the other hand, interpretivism seeks a rich understanding of the context so the researcher will pursue a hermeneutic phenomenological inclination to be ‘inside’ the research in order to *interpret* the meanings and describe his or her understanding of the context. Researchers who adopt a critical stance will seek to improve the social reality and as such will *intervene* in the context and generate change.

![Figure 6-3: Framework for Organizational multi-disciplinary research (Braa and Vidgen, 1999)](image-url)
6.3.4 Researcher’s epistemological stance

Each of the three forces seen in the framework presented in figure 6-3 above have an influence on the researcher’s preferences in relation to the general pursuit of knowledge. The applied vocational nature of ISD combined with the researcher’s industrial background in this field prompts a critical ‘interventionist’ mindset that his research be relevant and useful to the participants. The detailed ‘clean’ orderly deductive (and reductive) rigour of positivism is attractive to the algorithmic programming and design skills he has evolved through years of ISD application. However, it is challenging to determine situations within ISD where an objective reality can be controlled and replicated in a positivistic manner. Clearly examinations of the efficiency or reliability of technical information systems processing packages are candidates for positivistic research. In such cases, the reality being researched is restricted to an objective technological world of data, algorithms, network infrastructures, operating systems and hardware processing power. Such examinations are not to be dismissed as they are very relevant to the generation of product sizing guides for commercial off-the-shelf (COTS) products. However, they form only one small part of the large ISD research agenda and an even smaller part of the wider IS discipline. In this researcher’s opinion, ISD activities are essentially humanistic resulting in the need for most propositions in this area to consider some aspect of meaningful intentionality. The richness and potential complexity of this social reality informs this researcher that a core facet of any research in this field is the pursuit of understanding of the meanings applied by participants to their actions. So, the epistemological position held by this researcher is that in the pursuit of knowledge within social systems such as ISD organizations, an interpretive approach is desirable. In general his preference within such work would be to adopt a critical stance. However, in this particular study, he adopts a more benign observational presence so that he can effectively seek understanding of the situation. A major reason for this approach is to avoid the risks associated with researcher intervention in the community being examined. Finally, in deference to his attraction for the logical rigour of positivism, the researcher uses this structure and a broad reductive technique to support the construction of his understanding of the social reality via an orderly reasoning process.
In his exploration of published IS research, Walsham (1995a) uncovered four categories of justifications for the application of interpretivist approaches over the dominant positivist stance: “exploratory”, “complementary”, “appropriate to the topic” and “replacement of positivism”. Describing interpretive approaches as exploratory posits that such research may lead to the generation of hypotheses that may be later confirmed using positivist approaches. The complementary nature of interpretivism alludes to equality between both approaches and appreciation for the plurality of methodologies. Topic appropriateness claims that particular research areas are more suited to interpretivism (e.g. social science). The final category claims that positivism should be replaced by interpretivism. This researcher supports the third category by making the claim that an interpretivist stance is appropriate to the subject area under research.

The allowance of value-based interpretations recognizes a social reality constructed from interactions between humans that have applied their own meanings to their individually constructed realities. Without the opportunity to gather rich ‘facts’ and subsequently synthesize the patterns observed between members of the social system, it could prove problematic to address the complex nature of such systems. A deterministic predictable law or hypothesis that is confirmed via causality would need to be very specific and even then may not successfully cover the nuances of individual subjective meanings. Galliers and Land (1987) considered the difficulties in attempting to replicate reality in social science by statistically analysing particular factors and cautioned that this could result in misleading findings (cf. section 6.4.1).

Although this researcher has adopted an interpretivist epistemological stance, positivistic influences are evident in the application of this study. Walsham (1995a) stated that Lee’s (1991) approach to integrated research is applicable to IS. This approach proposed three levels of understanding in social research:

(v) the subjective understanding of the participants within the community
(vi) the interpretive understanding of the researcher as he/she analyses the social reality
(vii) the positivistic understanding created from formal testing in an objective way by the researcher.
He explained that Lee recommended that the logical disciplined approach espoused by positivism could complement interpretive understandings and help the researcher to predict and either confirm or refute alternative understandings. However, he highlighted that some interpretivists would consider the de-selection of alternative understandings to be problematic as it would dilute the overall ‘rich description’ of the situation. Also, the act of selecting or deselecting different understandings must involve a reliance by the researcher on their own understanding and as such runs contrary to value-free analysis dictated by positivism so such an approach could not be termed “positivistic understanding” (Walsham, 1995a). This explanation exposes an underlying fragility with positivistic understanding. However, it must be noted that awareness of a researcher’s understandings (or values) could be very helpful in using any knowledge produced from their interpretive understanding of a social reality. Rigorous analytical procedures may contribute to such awareness.

As stated earlier, this researcher does not attempt to integrate two epistemological stances that enable both a divergent exploratory examination of a phenomenon and a convergent deductive inspection of claims generated from that examination. However, there is an attraction to the notion of ‘borrowing’ the discipline of the hypothetico-deductive approach to structure the reasoning, analysis and generation of the interpretive understanding (or second-level construct as it has been described earlier). Lee and Baskerville (2003) observed a consistent recommendation from social scientists that while it is appropriate to record the subjective instances of individual participant realities by rich descriptions, additional rigour is required for analytic generalization. Second order constructs are on a different level and represent a theory that has been developed using the values of the researcher and as such are subject to the procedural rules of that researcher’s science. As stated earlier, positivism has held sway as the dominant approach within the IS research community. While reduction through rigorous causality may not be appropriate to social systems research, leveraging the broad analytic hypothetico-deductive structure may assist in the organization of reasoning and contribute to an understanding of the social reality. Braa and Vidgen (1999) noted that while reduction is associated with positivistic research methods, interpretivist research methods also must incorporate an element of reduction in order to assemble information considered relevant to a particular inquiry. However, they stated quite
clearly that “such a ‘reduction’ is not rationalized through the application of the systematic procedures of positivism” (Braa and Vidgen, 1999, p. 27). Although this researcher adopts an interpretive epistemological stance for the reasons outlined earlier, it seems reasonable to leverage a ‘positivistic-like’ analytic process in order to create the second level construct that forms the theory on the LSD/ASD relationship. As a result, the method that is selected here is an interpretive case study that enables the researcher to construct rich idiographic details of his interpretations of the intentions of the ISD team in performing particular Scrum practices. The in-depth discussion on research application and method characteristics presented in section 6.4 provides further discussion on the ‘soft/hard’ overlap afforded by the case study method.

In summary, the researcher has based his approach to this study on a phenomenological view of reality. This decries the notion of reality being an objective physical set of entities and embraces a reality that is composed of subjective meaningful entities that interact to construct a common social reality. As a result, the need to elicit knowledge from this reality requires mechanisms that enable value-laden interpretations to be described in a rich subjective manner and subsequently analysed according to the values of the researcher in order to generate an analytic generalization or theory about the phenomenon. In this case, it is to understand the manner in which Scrum ISD has been performed within globally distributed teams and to interpret potential associations between LSD and that performance.

6.4 The Methodological Question

This section addresses the selection of an appropriate method to conduct this interpretive research study. The section begins by presenting a general description of different methods and their suitability to support different epistemological stances. The rationale for selection of one particular method (soft case study using rigorous analytic procedures) is then explained. A high-level description of the case study design is presented and this is followed by an evaluation of the design using a framework that leverages principles of interpretive case study research.
6.4.1 Characteristics of methods

Galliers and Land (1987) expressed a concern over the “positivistic hegemony” observed by Fitzgerald and Howcroft (1998) that prevailed in IS research throughout the 1980’s. They noted that this situation led to IS being treated as a natural science and up to 50% of research efforts being expended on causality-based deductive approaches such as laboratory experiments and field surveys. Their concern with such approaches was that the messy real-world is not easily simulated in a laboratory experiment and in cases where it is difficult to apply values to variables, there is a tendency to exclude them from the analysis. The danger with this situation is that results may be misleading and could differ greatly if additional variables were present. This has led to the practice of disclaimers and recommendations for further research to refute or confirm the findings. They observed that such approaches often do not result in useful knowledge and reduce the relevance of IS research. As a result of the diversity of this particular field of inquiry (incorporating social sciences and business as well as natural science), it is important to leverage more appropriate research methods rather than blindly rely on traditional deductive approaches. They categorized IS research approaches as either being suited to empirical observations or supportive of in-depth interpretations. Galliers (1993) refined this categorization by presenting an overlap between observations and interpretations, placing case study and surveys much closer to the interpretive model.

The use of dichotomous characteristics to distinguish different aspects of research is applied in the discourse on IS research. In these dichotomous relationships, characteristics tend to align with a positivistic (objective/hard/predictable) or interpretive (subjective/soft/messy) orientation. They reside at different levels of abstraction: ‘ontological’, ‘epistemological’, ‘methodological’ or ‘axiological’. Table 6-1 presents a sample list of dichotomies at the methodological level. A notable contention proposed by the authors of this table is that there may be room for complementary application of opposing stances at the methodological level (Fitzgerald and Howcroft, 1998).
As an advocate of the pluralist application of multiple methods to address different dimensions of IS research, Mingers (2003) investigated the literature to determine how methods are applied. He classified methods according to their general suitability within three underlying paradigms although a caveat was declared that methods may be used across paradigms. Categories of methods that epitomize a positivistic or ‘hard’ approach include passive observation, questionnaire, experiments, simulation and case study. Methods associated with interpretive research include interviews, content analysis, ethnography/hermeneutics and grounded theory. The third category of methods (intervention) includes action-research and consultancy. Action-research involves intervention by the researcher using a particular theory to try and bring a change to the situation. On the other hand, consultancy is the provision of work for payment and may be used to gain knowledge about the domain being worked upon and the consultancy process undertaken(Mingers, 2003).
The three paradigms: positivist, interpretive and interventionist align with the outcomes put forward in the research framework of Braa and Vidgen (1999) in their categorization of methods to be used in order to seek prediction, understanding or change (figure 6-4).

The case study method is used where a phenomenon and its context are interleaved and it is not possible or useful to examine the phenomenon outside of the context. This method has been described as both positivistic (Yin, 2009; Runeson and Höst, 2009; Jansen and Brinkkemper, 2008; Eisenhardt, 1989; Mingers, 2003) and interpretive (Walsham, 1995b). Galliers (1993) positioned the case study approach within the positivistic umbrella but close to the interpretive end of the spectrum by acknowledging its proximity to observation-based methods. Braa and Vidgen (1999) acknowledged the debate by describing the case study approach as being applicable to both positivistic and interpretive preferences. They distinguished between ‘hard’ and ‘soft’ case studies although they appeared to align this method more closely with interpretivism (figure 6-4).

**Figure 6-4: Appropriate methods for different approaches (Adapted from Braa and Vidgen, 1999)**

The case study method is used where a phenomenon and its context are interleaved and it is not possible or useful to examine the phenomenon outside of the context. This method has been described as both positivistic (Yin, 2009; Runeson and Höst, 2009; Jansen and Brinkkemper, 2008; Eisenhardt, 1989; Mingers, 2003) and interpretive (Walsham, 1995b). Galliers (1993) positioned the case study approach within the positivistic umbrella but close to the interpretive end of the spectrum by acknowledging its proximity to observation-based methods. Braa and Vidgen (1999) acknowledged the debate by describing the case study approach as being applicable to both positivistic and interpretive preferences. They distinguished between ‘hard’ and ‘soft’ case studies although they appeared to align this method more closely with interpretivism (figure 6-4).
Ethnography refers to inquiry of a culture by the researcher’s deep involvement in their environment in order to interpret phenomena through the eyes of the subjects. Strict non-participation is termed ‘pure ethnography’ as against ‘participant observation’. Hermeneutics addresses aspects of this deep research that require the interpretation of textual material. Klein and Myers (1999) stated that there is little to distinguish ethnography from in-depth case studies other than the fact that ethnographers spend a greater length of time in the field.

6.4.2 Method selection

Controlled experiments may be overly restrictive in attempting to isolate and deduce particular findings associated with software development practices. The case study method enables researchers to build a rich picture of the context and situation under investigation. It is suited to many software engineering research initiatives as the collection and analysis of rich data may address the complex environment of integrated processes and roles engaged in ISD projects (Runeson and Höst, 2009).

In their presentation of guidelines to assist the development of case studies in software engineering, Runeson and Höst (2009) declared that their studies reflected a positivistic slant. A ‘soft’ or interpretive case study is considered more appropriate to the pursuit of meaning and understanding of a situation (Braa and Vidgen, 1999). An analysis of ISD method research found that case study investigations in this regard were mainly interpretive (Wynekoop and Russo, 1997).

Klein and Myers (1999) reflected upon three types of case study (positivist, interpretive and critical) and confirmed that the guidelines from the positivist stream have become the de facto standard for case study evaluation in IS research. However, they pointed out that using such criteria is inappropriate for the evaluation of interpretive research. As a result of this gap, they proposed a set of principles based upon hermeneutic philosophy that are intended to assist in the implementation of an interpretive case study research process. They accepted that some researchers will resist the notion of a guiding framework as it may “violate the emergent nature of interpretive research” while other researchers will consider it useful to the process (Klein and
They adopted a neutral position in this argument and declared that although they believed that the systematic application of criteria is not consistent with an interpretive approach, they felt that cognizance of a set of principles based upon interpretive philosophy is better than having no standards.

This researcher agrees with the view that interpretive research should not be evaluated based upon systematic deductive ‘rules’ as proposed by (Runeson and Höst, 2009; Yin, 2009). However, the position is taken that such structured rules are useful in ‘sense-making’ and interpretation of the idiographic data gathered from the case situation. The use of a case study protocol and associated commentary also serves to transparently communicate the interpretation process performed by the researcher, thus revealing insights on the researcher’s values based upon his background and history. This is a core element of interpretive research - the involvement of the researcher in the process. Therefore, in this study, the use of a hypothesis and associated proposition to structure the deductive analysis of a particular concept is not applied to falsify a causal law. It serves to bound an investigation and as such forms part of a conceptual framework related to the inquiry (Miles and Huberman, 1994). To further expound this point, it seems appropriate to describe and review an interpretive research approach that leverages a rigorous case study protocol using the aforementioned principles of interpretive field studies. Section 6.4.3 presents the high-level research approach. The methodological question concludes by using the principles of interpretive case study research to review the high-level design. This review contends that in this research study, the application of a systematic case study approach is consistent with the researcher’s interpretive epistemological stance.

6.4.3 High-level Design Approach

As stated in section 6.1.1, the core objective of this study is to explore the relationship between LSD and ASD. Tightening this objective further, the overarching research question to be answered inquires about the presence of LSD values in the application of Scrum teams within a GSD context. A two-phase approach is taken to the research process:
(i) **Conceptual framework development.** This involved hermeneutic analysis of literature related to lean software development and the application of Scrum in GSD. The product of this phase is a model based upon method rationale analysis theory which links LSD values to particular goals and the associated Scrum practices that are performed to achieve those goals. This model is called the ‘candidate LDScrum’ model and represents the conceptual framework used to scope the empirical research conducted in the next phase.

(ii) **Empirical Research.** This was an in-depth single case study of an ISD organization that leveraged the Scrum framework using globally distributed teams. Two projects from this case were explored. Each project deployed its team using a different configuration. The two configurations studied were ‘Distributed Scrum of Scrums’ and ‘Totally Integrated Scrum’ (cf. table 4-8). A detailed case study protocol was devised using the guidelines proposed by (Yin, 2009). Data was collected using project documentation databases and team member interviews. This data was analysed rigorously in order to gain a deeper understanding of the situation. The output of this work was a rich description of the situation and a ‘refined LDScrum’ model.

![Figure 6-5: Iterative research approach (Adapted from Yin, 2009)](image-url)
A set-up stage formed the basis for data collection and overall planning. It led to the development of a case study protocol and a conceptual framework. The purpose of these artefacts was to support the creation and execution of a structured research process. It defined clearly the type of data to be gathered, focussed the analysis and promoted consistency. This structured process helped to guide activities and avoid data overload (Miles and Huberman, 1994). This stage incorporated the first phase of the research design: conceptual framework construction. It also resulted in other detailed preparation for the empirical research. Yin’s initial approach recommended iterative feedback from stage 2 in order to refine both the study preparation and conceptual framework. This part of the approach was not followed as the candidate LDScrum framework remained as a consistent structure used to guide all data collection activities. Further details on the study preparation and theory development are presented in the next chapter.

The remaining two stages addressed data collection and analysis. The main mode of data collection for this research was interviews and development documentation. The data analysis process reflected the transformation of data to meaningful information and ultimately actionable knowledge (Turban et al., 2007). This involved reduction, display and verification activities. Reduction was achieved through the decomposition of data using coding techniques. Data display presented the coded data in a clear and manageable form so that it could be further processed. Finally, summarized data was segmented into different areas in order to determine patterns and conclusions related to different research questions and also to support verification of the analysis process.

### 6.4.4 Evaluation of the research approach

As stated above, in order to support the application of interpretive field studies, Klein and Myers (1999) proposed a set of principles that may be used by researchers to design, describe and assess their work. These principles are used to provide a high-level description of how this case study approach was applied. A key motivation for providing this description is to clearly explain the interpretive approach and present the reasons why the researcher has used what appears in the detailed design to be a positivistic application of the case study. The intention is to prevent
any confusion related to what might appear as a contradictory use of this method. A secondary motivation is to present an evaluation of the overall interpretive research work based upon these principles.

The first principle refers to the hermeneutic circle. Hermeneutics is considered to be a philosophical stance (in much the same way as earlier discussions on ontology and epistemology) or to be a particular analysis approach. It is mainly considered here as an analysis approach although the philosophy of hermeneutics informed the establishment of these principles. Hermeneutics is concerned with understanding the meaning of textual data (or any actions that may be presented as text - such as a conversation). A fundamental assumption of philosophical hermeneutics is that it deals with meanings. Words do not express facts - they express meanings. Table 6-2 presents each of the 7 principles of this interpretive framework and explains how the principle is applied in the research design. A brief description of each principle is outlined here:

(i) The fundamental concept of the hermeneutic circle proposes that development of understanding about an overall reality is achieved through preconceptions of the total reality and the relationships between parts of that reality. An iterative examination of these parts and reflection on the initial understanding of the overall situation can serve to deepen the overall understanding by confirming certain preconceptions and uncovering misunderstandings. At one level, this iterative approach may be applied to deepen the researcher’s knowledge of a particular area through the iterative interpretation of texts in order to arrive at a deeper understanding of the shared meanings presented by different authors in relation to a particular phenomenon.

(ii) The principle of contextualization highlights the need to acknowledge that different social and historical contexts result in different realities. This runs contrary to the positivist ‘rule-making’ approach to research that expects a scientific deduction of a situation to result in a rule that may be applied in future instances of that situation. In interpretive research the key is to make the context visible in order to promote
understanding of the situation. This supports the notion of generalizing a case by assisting future researchers in their understanding of the specific situation that is examined.

(iii) The third principle prompts the researcher to consider their interaction with the subjects and reflect upon concepts such as any preconceptions they might have about the subjects. Such assumptions may influence interpretation of observations and impact the analysis.

(iv) The fourth principle gets to the heart of the methodological dilemma encountered by this researcher. It discusses the concepts of abstraction and generalization. While acknowledging that interpretive research does not seek to test theory in order to refute causal laws, it is useful to use theory to explain the sensitization process. The linkage of interpretive research to a theory may assist in describing how the researcher made sense of the data. An orderly categorization of idiographic details using a theoretical foundation can help steer the data toward the relevant topics being examined. However, this researcher accepts that there is a risk associated with reliance on theoretical constructs. Such reliance may lead to a single-mindedness that could serve to conceal potential interpretations and associated insights.

The final three principles address concepts that may impact upon the interpreted outputs:

(v) The principle of ‘Dialogical Reasoning’ refers to the necessity for transparency in relation to the original research design. The researcher must present their beliefs in relation to ontology and epistemology and explain how these beliefs guided the research approach and implementation. This will highlight any prejudices that may be important to interpretations.

(vi) The principle of ‘Multiple Interpretations’ notes the need to consider that different subjects may provide different interpretations of a situation. Consideration of the reasons for differing perspectives may lead to greater understanding of the situation.
The final principle is most associated with critical theory and relates to the need for the researcher to be suspicious of deliberate distortions or untruths in the data due to various forces at work within the situation.
<table>
<thead>
<tr>
<th>Principle</th>
<th>Application in research design</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. The Fundamental Principle of the Hermeneutic Circle</strong>&lt;br&gt;&lt;i&gt;This principle suggests that all human understanding is achieved by iterating between considering the interdependent meaning of parts and the whole that they form. This principle of human understanding is fundamental to all the other principles.&lt;/i&gt;</td>
<td>The research design embraced this principle in both the high-level structure of the design and the detailed application of aspects of the work. At a high-level, the ‘whole’ relationship between LSD and ASD is presented in a candidate LDScrum model. Constituent parts of this model are then empirically examined before raising the understanding back up to a new ‘whole’: the refined LDScrum model. Low-level application of this principle is evident in multiple parts of the study. For example, understanding the association between Scrum practices and the goals that motivate their execution was achieved by an initial establishment of 12 core Scrum practices (that represent the ‘whole’). A hermeneutic textual analysis of multiple sources examined each of these practices and induced supplementary practices that may be performed to achieve the practice. Following the examination and proposed association of each of these supplementary practices to various goals, they are then synthesized back to the core practices, a process which represents the journey from whole to constituent parts back to whole.</td>
</tr>
<tr>
<td><strong>2. The Principle of Contextualization</strong>&lt;br&gt;&lt;i&gt;Requires critical reflection of the social and historical background of the research setting, so that the intended audience can see how the current situation under investigation emerged.&lt;/i&gt;</td>
<td>Initial high-level management meetings were held in order to fully understand both the overall history of Scrum use within the organization and any major issues that have impacted the projects to be studied. This high-level management view serves to present a broad context that sets the scene for data collection from developers who may or may not have been part of the evolution of development approaches within the organization.</td>
</tr>
<tr>
<td><strong>3. The Principle of Interaction Between the Researchers and the Subjects</strong>&lt;br&gt;&lt;i&gt;Requires critical reflection on how the research materials (or ‘data’) were socially constructed through the interaction between the researchers and participants.&lt;/i&gt;</td>
<td>An air of formality was introduced to the process in order to preserve researcher neutrality and ensure consistency in data collection from former colleagues of the researcher as against subjects that were unknown to the researcher. It is acknowledged that researcher interpretations could be influenced by former relationships, especially in semi-structured interviews where clarifications of situations may be performed by reference to past shared experiences. Another aspect of this principle was the decision to refrain from any interventionist tendencies in order to retain strict neutrality. This point is addressed further in the fifth principle.</td>
</tr>
<tr>
<td>Principle</td>
<td>Application in research design</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td><strong>4. The Principle of Abstraction and Generalization</strong>&lt;br&gt;Requires relating the idiographic details revealed by the data interpretation through the application of principles one and two to theoretical, general concepts that describe the nature of human understanding and social action.</td>
<td>Lee and Baskerville (2003) referred to generalization of idiographic details to theory thus distinguishing the interpretive research from mere anecdotes. In this study, the LDScrum model leveraged method rationale analysis theory in order to devise a conceptual framework that served to structure the analysis of rich information provided through idiographic data.</td>
</tr>
<tr>
<td><strong>5. The Principle of Dialogical Reasoning</strong>&lt;br&gt;Requires sensitivity to possible contradictions between the theoretical preconceptions guiding the research design and actual findings (‘the story which the data tell’) with subsequent cycles of revision.</td>
<td>Chapter 6 presents the ontological and epistemological positions held by the researcher and uses these stances to explain the rationale for selection of the case study method. An observation related to this point is the author’s admission of his preference for critical thinking. This particular preference was suspended throughout the interpretive empirical research process in order to preserve a strict neutral observational position. This was important in this research, not only because it served to scope the work to be done, but also because it was important to the organization that their employees were not confused by feedback from the research.</td>
</tr>
<tr>
<td><strong>6. The Principle of Multiple Interpretations</strong>&lt;br&gt;Requires sensitivity to possible differences in interpretations among the participants as are typically expressed in multiple narratives or stories of the same sequence of events under study. Similar to multiple witness accounts even if all tell it as they saw it.</td>
<td>The use of coding practices within a rigorous analytic process enabled the researcher to compare multiple views on a practice or topic. Individual researcher interpretations of each of the multiple views could then be compared in order to arrive at various insights related to that practice within the social reality. Different viewpoints may then be highlighted in the case report with potential reasons for the differences. One developer may feel that story points are an effective tool whereas another may not. A potential reason for this disagreement may be identified as developer experience.</td>
</tr>
<tr>
<td><strong>7. The Principle of Suspicion</strong>&lt;br&gt;Requires sensitivity to possible ‘biases’ and systematic ‘distortions’ in the narratives collected from the participants.</td>
<td>The open-ended iterative interview process enables clarifications to be sought where a distortion emerges. Data reduction displays using common codes may also reveal a different interpretation that could be a deliberate distortion. In certain cases, developer documentation may help to clarify the truth of the situation.</td>
</tr>
</tbody>
</table>

Table 6-2: Description of overall interpretive research - adapted from (Klein and Myers, 1999)
6.5 Summary

This chapter opened by distinguishing between natural science and social science. It explained that investigations in natural science favour researcher independence and the deterministic development of laws as a result of logical derivations from causal relationships. Conversely, the study of social science is more suited to rich descriptions and researcher interpretation of community members’ judgements and perceptions. This particular study is engaged in IS research, a scientific field that is quite diverse and fragmented. In order to further examine the research approach, the subjects of ontology and epistemology were discussed in both general terms and also in the context of this inquiry into ISD. The dichotomy of realism and idealism were described and the researcher declared his ontological position to be a belief that an inter-subjective reality exists between a social community. A discussion on epistemology explained how this topic is dominated by the choice between a preference towards logical deductive derivation of knowledge (positivism) or rich descriptive interpretive revelation of knowledge (interpretivism). A description of the evolution of scientific research revealed a recurring concern over the application of closed deductive reasoning to social science. Positivistic and interpretive epistemological stances were reviewed from both general and IS-specific research perspectives. Arising from this discussion, a detailed explanation was provided describing this researcher’s views on positivism and interpretivism. This explanation presented the reasons for his selection of an interpretive stance and also his inclination to use a rigorous analytical structure to support the application of this approach. The methodological question reviewed different research approaches and proposed that a case study approach is most appropriate to meet the research needs. A high-level view of this approach was outlined and the chapter concluded by using a series of principles to describe the high-level interpretive research design.
Chapter 7- Research Detailed Design

7.1 Introduction

Figure 7-1 highlights the WBS segment responsible for the project scope related to research design activities. Chapter 6 described the first work package in this area: ‘2.1.1’. Work packages ‘2.1.2’ and ‘2.1.3’ address detailed research design considerations. This chapter describes ‘2.1.2’ whereas the case study protocol (2.1.3) is presented in appendix B. A more detailed view of the entire WBS is available in figure 1-7.

![Figure 7-1: Research approach WBS segment (detailed design & case context work packages)](image)

This chapter explains the detailed design decisions and activities that guided this interpretive case study. Given that the research strategy determined in chapter 6 has selected a preference for an open-ended idiographic value-laden research approach, it is deemed important that this chapter presents detailed contextual information on both the researcher and the environment studied. To this end, the chapter opens with a discussion on the management of certain concerns.
in relation to interpretive research fieldwork and explains why the situation under investigation is considered to be typical of ASD in GSD. The body of the chapter (sections 7.2 and 7.3) present details on the design decisions taken to manage the investigation. The approach described is primarily directed by insights gained from three sources: (Miles and Huberman, 1994; Yin, 2009; Runeson and Höst, 2009). Although the latter two references here are associated with positivistic case study research, their guidelines are adopted in order to support a consistent data collection and analysis structure. As stated in chapter 6, the primary goal of this research design is to present a rich idiographic picture that serves to reveal insights about the relationship between LSD and ASD. Detailed frameworks, research questions, units of analysis and other research design artefacts are used to support this goal. A secondary objective declared in chapter 1 is to support future research efforts into this topic. These research design artefacts serve to support transparency of the work performed and may prove useful to future studies in this area. Finally, in order to further support the interpretive nature of this work, the chapter concludes with a description of the context of the organization under examination.

### 7.2 Major design influences

Walsham (2006) noted two concerns that influence interpretive fieldwork: researcher involvement and data access management.

#### 7.2.1 Fieldwork: Researcher involvement level

Walsham (2006) suggested that the researcher needs to determine their level of involvement in the study. He considered involvement as a ‘spectrum’ of states ranging from a researcher being perceived as independent from the organization being studied through to a fully engaged action researcher who is attempting to apply critical thinking in order to enact change within the organization. The advantages of heavy involvement include access to rich data as participation in activities may often provide more detailed exposure to a situation than second-hand interpretations of opinions. It can prompt greater cooperation as the relevance of the research may be more evident to the research subjects. However, ethnographic or action research study is time-consuming and poses certain risks. Subjects may be reticent to share honestly their views
with inquiries that they may perceive as contributing to a vested interest. Another potential risk to the heavily involved researcher is the loss of perspective on the theory and becoming embroiled in the tactical issues facing the organization (Walsham, 2006).

Care was taken with the preparation and execution of this research to manage the participant’s perceptions in relation to the level of involvement of the researcher. The objective in managing this involvement was to gain advantage by obtaining deep access to the materials under study while also overcoming any potential disadvantage due to a perceived lack of neutrality. At different times in the set-up of fieldwork activities, the point was reinforced that this work was being performed to reveal insights on an ISD situation and not to transform the social reality being examined. This point was made at initial management meetings and at the introduction of each field interview in order to preserve the researcher’s ‘neutrality’ in the eyes of participants and alleviate any concerns that they may have held in relation to his intentions. As a former employee of the organisation under inquiry, the researcher was in a position to present his plans to senior management and get support for his work. This posed a serious risk to the researcher’s neutrality. A strategy was pursued to maintain ‘distance’ between the researcher and the organization while at the same time embedding him in their activities. The researcher submitted a proposal for a formal collaborative research relationship between the organization and a national research centre. Then the researcher presented his work under the umbrella of a research project, thus reducing his role to one of many potential collaborators. Furthermore, an air of formality was upheld in the arrangement of all meetings and field interviews in order to maintain a professional distance as some of the participants had been former colleagues of the researcher. A consistent formal data collection protocol was followed to ensure no distinctions were made between engagements with former colleagues and participants that were unknown to the researcher.

---

10 LERO - The Irish Centre for Software Engineering Research is a centre for science, engineering & technology (CSET) supported by Science Foundation Ireland (SFI) (www.sfi.ie)
11 ALPS (Agile and Lean Project Management and Scaling) project is part of the LERO CSET research program
7.2.2 Fieldwork: Data Access Management

Another consideration in addressing interpretive research is how to gain access to useful research subjects and subsequently keep them engaged in the initiative. The researcher’s previous employer was deemed a useful research subject due to their strong experience in GSD and their use of the Scrum ASD method. Access to this organization was carefully planned and involved emails, phone meetings and a face-to-face lunch meeting with the organization’s US-based Chief Technical Officer in order to get high-level awareness and approval for the study. One individual was identified and established as a central liaison to support maintenance of the collaborative momentum over the relatively long-term duration of the PhD project. The aforementioned formal collaborative agreement further cemented the relationship. It also served both to position the researcher as part of a wider initiative and to support ongoing feedback activities from the research to the organization.

The provision of research feedback supports the collaborative relationship as the research subject perceives benefit from their involvement. Due to their interactive nature, workshops can be very effective in this regard (Walsham, 2006). As a formal partner of the ALPS project, the organization participated in multiple workshops related to LSD and ASD issues. Other forms of feedback included emails and regular liaison status meetings. Management of fieldwork concerns in this study are summarized in table 7-1.

<table>
<thead>
<tr>
<th>Fieldwork Issue</th>
<th>Concerns</th>
<th>Solution applied in this study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Researcher Independence</td>
<td>May reduce richness of data access. May avoid bias through over-involvement in work.</td>
<td>Presented as independent researcher from LERO. Formal interview process. However, given employee privileges enabling rich data access.</td>
</tr>
<tr>
<td>Gaining access to research subjects</td>
<td>Difficult to gain access to appropriate research participants and data</td>
<td>Project management approach to data access. Obtained CTO support for collaboration and established clear management liaison to support data access activities.</td>
</tr>
<tr>
<td>Maintenance of research access</td>
<td>Need to maintain collaborative relationship through regular feedback to study participants</td>
<td>Conducted workshops and regular status meetings with organization</td>
</tr>
</tbody>
</table>

Table 7-1: Summary of solutions to fieldwork issues reported in Walsham (2006)
7.2.3  Design approach: Single v Multiple Case Study

“What, for example, is wrong with samples of one? ...Given that we have one hundred people ...we should ask ourselves whether we are better off to have each study 100 organizations, giving us superficial data on ten thousand, or each study one, giving us in-depth data on one hundred...”  
(Mintzberg, 1979, p. 583)

A single case study design is adopted for this research as the situation being investigated is deemed to be a typical instance of ASD method application within a GSD environment. The situation is considered typical from both ASD and GSD perspectives:

(i)  **ASD.** The organization is a mature R&D division of a COTS product development company currently migrating from a ‘waterfall’ ISD to Scrum. Scrum is considered the most widely-used ASD method. Given that most professional ISD methods are based upon the traditional software development life cycle (Fitzgerald et al., 2002), it seems reasonable to state that ISD migration to the most dominant ASD method from the most dominant ISD approach would be a typical representation of ASD adoption within mature COTS product development organizations.

(ii) **GSD.** The organization initially leveraged GSD in order to pursue decreased labour costs and enable a longer global working day by extending their core R&D operations from California, USA to Limerick, Ireland and Sydney, Australia. The R&D organization has been further dispersed by the addition of software development teams through acquisition of different product development organizations. Decreased labour costs, extended working days and company acquisitions are all cited as typical motivations for organizations to pursue GSD (Carmel, 1999).

7.3  Detailed Design Stage 1 - Study setup

The three stages that form the research design (‘Study setup’, ‘Data collection’ and ‘Data analysis’) were introduced in chapter 6 (figure 6-5). This section reviews the two main activities performed to setup the detailed design. These activities were the establishment of research
management components and the construction of a conceptual framework. The decisions taken to define each of the management components are described. This is followed by a description of how development of the framework required a structured examination of LSD, NPD, ASD and GSD considerations.

### 7.3.1 Management of Research Components

There are five basic components of case study research design (Yin 1994):

- (i) **Research questions**
- (ii) **Research propositions**
- (iii) **Units of Analysis**
- (iv) **Data analysis approach: linking data to the propositions**
- (v) **Criteria for processing the findings.**

#### (i) Questions and (ii) Research propositions

The overall remit of this inquiry is to explore the relationship between LSD and ASD or more specifically, LSD and one variant of ASD - the application of Scrum in GSD. Runeson and Höst (2009) adapted Yin’s case study guidelines in their research into software engineering practices. They established a set of research questions which they decomposed into propositions that led to hypotheses to be examined by subsequent activities. This positivistic approach is used as a regulating structure to support analysis of the interpretive research. Each hypothesis is merely a holder for one or more propositions which in turn are just mechanisms to group findings into analysis sets and keep the research relevant to its original remit. Four research questions (RQ1 - RQ4) are presented. A research question may be used to generate one or more propositions (e.g. PR1.1). Each proposition is defined as a hypothesis to be examined (H1.1). Table 7-2 presents the application of this structure to the case under investigation. These hypotheses are generated as a result of the linkages proposed in the conceptual framework constructed from analysis of the literature (chapters 2, 3, 4 and 5). This analysis indicated the possibility that Scrum performed in GSD may exhibit NPD characteristics, achieve ASD goals and overcome GSD challenges. The
framework also proposes associations between LSD values and the pursuit of goals related to those NPD, ASD and GSD concepts.

RQ1: How do distributed Scrum projects relate to the characteristics of successful new product development teams as proposed by (Takeuchi and Nonaka, 1986)

PR1.1: \textit{IF a distributed team performs the ISD method, SCRUM, THEN they will be seen to exhibit the characteristics of a successful new product development team}

H1.1: A distributed team that uses Scrum will exhibit all 6 characteristics of a successful new product development team (Built-in Instability, Self-organization, Subtle control etc.)

RQ2: How do distributed Scrum projects address the principles of Agile software development?

PR2.1: \textit{IF a distributed team performs the ISD method, SCRUM, THEN they will address the principles of agile software development}

H2.1: A distributed team that uses Scrum will achieve all 12 goals associated with the principles of ASD

RQ3: How do distributed Scrum projects overcome the challenges that face global software development (GSD)

PR3.1: \textit{IF a distributed team performs the ISD method, SCRUM, THEN they will overcome many challenges of GSD}

H3.1: A distributed team that uses Scrum will overcome the 3 identified GSD challenges

RQ4: How do distributed Scrum projects relate to lean software development values?

PR4.1: \textit{IF a distributed team performs the ISD method, SCRUM, THEN they will comply with lean software development values}

H4.1: The presence of all 12 identified Lean Software Development (LSD) values will be evident in projects that are performed by distributed teams that use Scrum

<table>
<thead>
<tr>
<th>(iii) Units of Analysis</th>
</tr>
</thead>
</table>

The main unit of analysis is the software development project. For each unit of analysis, three additional sub-units are examined:

(i) The ISD team. This unit of analysis is used to identify the team configuration. Each software development project is performed by globally distributed teams. The two team configurations analysed are ‘Scrum of Scrums’ and ‘Totally Integrated Scrum’.

| Table 7-2: Requirements, propositions and hypotheses of this case study |
(ii) The software developer. The Scrum framework recommends three roles: Product owner (PO), Scrum-master and Developer. Understanding the specific role that supplied particular interpretations may reveal interesting patterns. Furthermore, although the developer role is considered by the framework to be a generic interchangeable role, it is normal practice that developers have a primary skill or role. This role may be as a quality assurance specialist (QA), business analyst (BA) or a software engineer (SE). Information regarding the specific ‘developer type’ as a source is also collected in order to support this unit of analysis.

(iii) The ‘method fragment’ or sets of Scrum practices performed. (It is proposed from theory that Scrum consists of 12 core practices and these have been synthesized into four sets: general considerations, pre-sprint, sprint and post-sprint.)

(iv) **Data Analysis approach**

The data was linked to the propositions by using a series of codes. A starting set of codes was generated from the initial conceptual framework (LDScrum). Incidents where practices were influenced by goals are coded to indicate the practice performed, the role performing the practice, the project, the team and the method fragment that contains the practice. These codes may be found in the case study protocol (Appendix B). In effect, the data was linked via the codes to the conceptual framework.

(v) **Findings processing**

Findings were interpreted to determine associations between Scrum practices and each of the three types of goals. In order to structure this analysis process, the findings were categorised within hypotheses representing associations to each type of goal: NPD, ASD and GSD challenge alleviation goals. This interpretation was performed in each case by the identification of the influence of these goals on the Scrum practices performed, supplemented by a subjective assessment of their impact and educated interpretation of their association to LSD values. The coding structure meant that the data could be filtered and sorted from different perspectives in order to generate a comprehensive multi-faceted view of the social reality. Rich insights may be

---

12 Note that hypotheses examination is incidental to this research - it was merely a device to focus the interpretive process and is not intended to be put forward as evidence of some causal law.
gleaned from this. These findings may be used to refine the candidate LDScrum model and may prove useful for future related research.

7.3.2 Theory Development

The conceptual framework served to guide the research process. The framework illustrated the interconnections between the three primary areas used in the investigation: LSD values, Scrum in GSD goals and Scrum practices. Detailed outputs from this portion of the research are presented in chapters 2, 3 and 4. The underlying theories and structures used to assemble the conceptual framework are explained in chapter 5.

Theory development involved hermeneutic analysis of literature related to LSD and the application of Scrum in GSD. The product of this work was a model based upon method rationale analysis theory which links LSD values to particular goals and the associated Scrum practices that are performed to achieve those goals. This model is called the ‘candidate LDScrum’ model and represents the conceptual framework used to support the empirical research. Construction of this model involved both a definition and an association process. The definition process required identification and declaration of the three sets of phenomena to be linked together: LSD values, Distributed Scrum goals and Scrum practices. The association process involved establishing linkages between each LSD value to particular goals and also association between each goal to particular practices.

(i) Definition of values, goals and practices. Definition of the LSD value set was achieved through a genealogical review of literature on the application of lean concepts to various disciplines ranging from their introduction in the field of manufacturing through to their more recent application in ISD. This review resulted in the identification of many sets of lean principles reported in various publications. A bottom-up approach was adopted in order to define a set of LSD values. A hermeneutic interpretive process was followed to synthesize a compilation of these principles into a set of lean values that can be leveraged to explore the relationship between LSD and ASD. The iterative synthesis of the values involved both convergence from narrow to broad (as similar values were grouped to a
higher-order value) and in some cases divergence from high-level to specific. Clearly there are limitations with this approach. The identification of values was subject to the bias of the authors’ beliefs and experiences. However, it is hoped that these beliefs and experiences are grounded in an appreciation of software engineering research and application and that the proposed value set resonates with members of these communities. This research is not intended to present the definitive set of lean software development values. It is intended to present a candidate set that may be used to facilitate further research into the relationship between LSD and ASD. The definition of distributed Scrum goals was established through the combination of three sets of goals related to NPD characteristics, ASD principles and GSD challenges. NPD characteristics were gleaned from the publication cited as the inspiration for the Scrum framework and as such deemed a likely source for the goals of Scrum practices (Takeuchi and Nonaka, 1986). A second source identified was the twelve principles of ASD as specified by the agile manifesto. The use of this source of ASD goals builds upon previous work on this topic conducted by (Ågerfalk and Fitzgerald, 2006; Ågerfalk, 2006). Finally, in order to establish a set of goals associated with ISD in GSD, a series of challenges associated with GSD (Ågerfalk et al., 2005; Holmstrom, 2006; Lane and Ågerfalk, 2008) are identified and goals established to alleviate those challenges. A set of ‘core’ Scrum practices are identified as a result of analysis of published reviews of Scrum from its first introduction by Schwaber (1995), through a major ASD contribution by Highsmith (2002) and up to a recent definition of the framework by its founders (Sutherland and Schwaber, 2011).

Association of values, goals and practices. In order to probe further into the Scrum method, additional detail on the application of the identified twelve core practices is supplied by examining contributions from various sources. Many additional Scrum practices are derived from an iterative analysis process of practice sets gleaned from papers and books that examine the Scrum framework. The analysis approach categorised practices as addressing general considerations associated with Scrum or being associated with different phases of a Scrum project lifecycle. All of the supplementary practices are synthesized into the core practice they support and associated goals of supplementary practices are associated with the ‘parent’ core practice. The results of the synthesis of
these practice-goal associations are presented in table 3-11 (core Scrum) and table 4-10 (Scrum in GSD). Chapter 5 (section 5.3.2) outlines how the combination of the synthesized mappings is used to create the DScrum model which presents a hierarchy of goals and associated practices that may be performed to achieve those goals. An iterative hermeneutic process was followed to associate LSD values to the DScrum goals. Descriptions of both LSD values and the LSD principles used to induce them are leveraged to inform this mapping. (cf. table 2-1 and table 2-3). These mappings are used to extend the DScrum model to include LSD values, resulting in the candidate LDScrum model. In order to simplify presentation and application of the overall model, it is segmented into three method fragments: ‘pre-sprint’, ‘sprint’ and ‘post-sprint’.

7.4 Detailed Design Stage 2 - Data Collection

Initial management interviews were conducted in order to increase understanding of the case, and plan data collection activities. This included the identification of the optimal candidate units of analysis (software projects) and the coordination of interviews with the managers of those projects. Those subsequent management interviews were used to develop the specific context description for each project investigated and support data collection activities within that project team. These activities included management approval to engage with the research and scheduling of individual interviews. In order to facilitate a comprehensive and rich exploration, all team members were interviewed. The four characteristics of the focused interview as defined by Merton and Kendall (1946) were present in all of the interviews:

(i) The situation (Scrum in GSD) had been analysed resulting in a set of propositions to be examined.
(ii) A set of semi-structured questions had been generated from the analysis.
(iii) Interviewees were selected based upon the fact that they had been involved in the application of Scrum in GSD situations.
(iv) Interviewees were examined on their views of the impacts of using Scrum in the GSD situations under discussion.
Each focused interview was approximately 100 minutes in duration and was directed by the research questions. However, candidates were allowed latitude to explore any aspects of an incident that emerged from the discussions. All interviews were recorded to allow the conversation to flow. This also reduced the risk of data recording errors. Transcripts were generated post-interview and digital audio recordings facilitated alteration of playback speeds to support the transcription process. Each interviewee was advised that the transcript was available for review. No further vetting of transcripts was performed.

Merton and Kendall (1946) recommended that prior analysis of an interview situation is a prerequisite of the focused interview. Development of the conceptual framework combined with the aforementioned senior management meetings enabled a clear analysis of the situation. The interviewer was familiar with the development environment as he had worked there as both a software engineer and manager. Familiarity with the situation is considered positive as it can lead to an effective interview process. However, care needed to be taken to ensure that leading questions were avoided and that the views of the interviewee were represented. In order to achieve this, the researcher adopted a semi-formal approach to engagement, using interviewee invitations and pre-delivered documentation to introduce the process. Interviewer prompts were limited to confirmation of recall issues or replies to interviewee requests. It was hoped that this formality would preserve researcher neutrality and maintain some distance between the parties. It was intended that this would ensure that the interviewee reviewed incidents from their own perspective and judgement (Merton and Kendall 1946).

An initial pilot interview was conducted leading to modifications to the interview protocol. An iterative process was used so that any insights gained from data collection and transcript production of an interview were used to improve the interview protocol for subsequent data collection activities. The researcher was granted full employee access to all developer documentation. This documentation was used to support both data collection and analysis activities.
7.5 Detailed Design Stage 3 - Data Analysis

As described earlier in chapter 6 (section 6.4.3), analysis involved three steps to reduce the data, analyse the partitioned data and propose conclusions arising from its review. Data reduction was performed separately for data produced from each interviewee. However, analysis and conclusions required an integrated view to analyse data using views across the units of analysis. In order to manage this data, a documentation process was created. Figure 7-2 presents an overview of the process and documents produced to manage these steps.

![Diagram of Data Analysis process]

Figure 7-2: Data Analysis process
7.5.1 Data Reduction

The artefacts generated during data reduction were contact summary sheets, coded transcript printouts, coded transcript softcopies (MS-Word) and a findings data dictionary (MS-Excel worksheet). Early data reduction efforts resulted in updates to the initial code list and the modification of interview questions. These modifications were prompted by themes that emerged from contact summaries and also insights gained while coding the raw transcript. An excerpt of the findings data dictionary is presented in appendix E. The techniques applied to develop each of these artefacts are outlined here.

(i) Contact summary sheets. A contact summary sheet was generated for each interview, capturing any salient points that emerged from that exercise. Each point was characterized by a theme or aspect, reflecting the fragment and goals pursued.

(ii) Transcript coding. An initial code list generated from the conceptual framework was used to process the interview transcript. A key attribute in all codes on the initial list denoted a particular goal that may have motivated the Scrum practice being described. Notable text excerpts were highlighted on a printout of the transcript and these excerpts were manually coded in hand-written side-notes to reflect researcher interpretations/observations related to various code assignments. The main interpretation noted here is the researcher’s views of why he thought a particular goal or goals were the motivation for the practice.

(iii) A secondary analysis was conducted using a softcopy version of the transcript. During this work, any observations by the analyst were recorded as memos. Microsoft Word’s text highlighting functionality mimicked the hardcopy highlighter and its comment functionality was used to record code assignments and related analyst’s observations. The secondary coding process enabled a deeper analysis of the data as the researcher confirmed his understanding (and coding) of a particular textual excerpt. He then reflected upon his interpretation of the subject’s interpretation in order to posit whether
an LSD value may have underpinned the goal. A key artefact used to support this interpretation was the descriptions of each LSD value (cf. table 2-3).

(iv) Findings ‘data dictionary’. The creation of a softcopy version of the coded and annotated transcript facilitated the transfer of all excerpts and related code/memos to other software applications as needed. In this case, each particular excerpt was copied to an Excel workbook. A sheet on that workbook contained all the coded excerpts with any associated memos. This sheet represented a data dictionary for research findings. This facilitated more rapid access and collation of coded excerpts for subsequent analysis.

7.5.2 Data Display

Data display and integrated analysis activities were all performed using MS-Excel. As stated above, a central worksheet contained a data dictionary of all data excerpts and associated codes and memos. The main unit of analysis is the software project. Each project studied generated a separate Excel sheet for each hypothesis (or category of findings). Excel’s data management facility enabled filtration of the data using codes. Each hypothesis sheet captured a summarization of the findings in relation to that hypothesis. This overall view set the tone for reflective analysis of sets of data in order to generate rich insights into the situation. It also prompted inspections of developer documentation from the organizations’ development documentation databases. These insights are intended to form the basis for the case report. The production of a report for each project was considered useful from both a research and practitioner perspective. As a research artefact, it serves to decompose the analysis and help to make sense of the final integrated case report. For practitioners, it is practical that reflections on their work are categorised by project as this is how the research might best relate to their working environment. Within each project, the summary hypothesis worksheet is finally updated with the researcher’s judgement in relation to the hypothesis. This was not intended to refute a theory - it just served to capture the ‘whole’ and complete the hermeneutic circle that represented the journey from the researcher’s original conception of the situation, through rich deep examination of different constituent parts of the situation resulting in a final overall perspective. This overall process is illustrated in figure 7-3 and described briefly below.
7.5.3 Conclusion and verification

The conceptual framework (or candidate LDScrum model) formed the initial ‘whole view’ and commences the ‘outer hermeneutic circle’. It informed the creation of each hypothesis and associated worksheet. Detailed collection and analysis resulted in empirical findings. Statistics from these findings were used to populate a hypothesis worksheet. These statistics were not used dogmatically to declare the researcher’s view on the hypothesis - they acted as a support to the interpretive process. As such, they served as a second starting point or ‘whole’ view (thus commencing an ‘inner hermeneutic circle’). The researcher reflected on different views of the data, referring to project documentation to help enrich his interpretations before finally returning to the ‘whole’ (the hypothesis summary sheet) and declaring his support or denial of the hypothesis (thus completing the ‘inner hermeneutic circle’). This process was performed for each
hypothesis within a project and resulted in a project report. In this research design, the facility exists to synthesize the findings further using the ISD team unit of analysis to identify the distribution structure used by the team. However, this was unnecessary in this particular study as a one-to-one relationship existed between the team configuration and project. (It could prove useful in future recommended applications of this framework as outlined in chapter 10). Once all the research was complete, a two chapter case report integrated all the findings. This case report was then used to modify the initial LDScrum model, thus creating the refined LDScrum model and completing the outer hermeneutic circle.

7.6 Reflections on design approach

The establishment of theory is recommended as a basis for the research design. This was interpreted by the researcher as equivalent to the use of a comprehensive conceptual framework. While not quite as ‘hard’ or logical as a scientific theory associated with natural science approaches, the conceptual framework proved very useful in establishing project scope. It helped to set boundaries and identify core areas that would need to be researched in order to establish mechanisms for data elicitation and analysis.

Iterative processes are highly recommended as a good practice from a high-level design perspective. (Yin, 2009) noted that a danger with iterative designs is that the researcher may lose their focus, especially if performing iterative explanation-building. Whereas this study is descriptive in nature, it was important that execution of the work remained relevant to the original question. As a past project manager, the researcher had always intended to pursue this research study using the best practices of project management (PMI, 2008), and as such it was critical to use a ‘charter’ and ‘work breakdown structure’ to define the work and establish the essential rationale and scope for the study. Both of these considerations were defined and managed within a key document recommended by Yin: the case study protocol (cf. appendix B). The protocol describes in detail the steps to be followed in the collection and analysis of data. All documents are outlined and the mechanisms used to produce findings are detailed. Adoption of an iterative approach meant that work on one major unit of analysis (ISD project) was completed
prior to the start of data collection related to another project. This resulted in a large amount of elapsed time between two executions of the same phase. The case study protocol reduced a lot of the effort in executing a phase. This was especially important for data analysis, as there were many steps and forms to be completed. An important feature in managing the research was the design and use of a case study database. The database promoted consistency and enabled traceability within the study. Its structure is outlined in the case study protocol.

Certain analysis techniques proved very effective. A clear coding structure that used a ‘supercode/subcode’ format of well-structured meaningful codenames aided recall and general reflection on the data. Memos applied to various data excerpts throughout the research process helped to preserve the history of the researcher’s views. This proved particularly effective for data reflection and production of the final case report. Use of a spreadsheet-based framework had many practical benefits. Filtering and sorting mechanisms permitted different views of the data, facilitating pattern matching and compilation of final findings. Formulae and cell-linking functionality facilitated the creation of a simple framework to manage case findings.

7.7 Case Background

“In order to judge whether a specific solution can be successful it is necessary to describe the context as completely and accurately as possible for the considered object of study.”

(Petersen and Wohlin, 2009, p. 401)

Petersen and Wohlin (2009) conducted a focused study on published industrial software engineering case studies, revealing that background information presented in such studies has been limited in its coverage of different aspects of the case study context. Aspects of study situations or ‘context facets’ were proposed as a checklist that may be used to guide and assess the description of case studies. A particular object of study (such as a Scrum team engaged in GSD) may be described in the context of four aspects: ‘people’, ‘processes’, ‘products’, and ‘practices, tools/techniques used’. These ‘inner-level’ facets are complemented by a wider perspective which consists of descriptions of the overall ‘organization’ and the ‘markets’ in which it operates. Figure 7-4 presents an adaptation of this model as it applies to the case being
studied in this initiative. The checklist and model are not exhaustive but are a list of contextual considerations to be included or discounted depending upon the situation. It is important that the presentation of context is not too detailed or too abstract and to support this objective, researchers are recommended to consider the aforementioned checklist when constructing case descriptions (Jalali and Wohlin, 2011).

In the spirit of the above recommendation, this chapter concludes by describing the case under investigation. Although the various context facets within the checklist guide the author’s presentation of the case, the description does not descend into a ‘join-the-dots’ table of facts and figures. It is considered more appropriate at this point to present a rich narrative on the case and refer to context facets as deemed appropriate. This section presents a high-level contextual overview of the case, using the two outer-level context facets to provide background on the organization under research. It is also considered relevant to provide an outline of the application of Scrum within the wider organization as this helps to position the two projects that were studied in detail. The presentation of findings from each of these projects is supported by a detailed context description of the project that leverages the four inner context facets (cf. chapters 8 and 9). It is hoped that the provision of this comprehensive contextual information will meet the aspiration of Petersen and Wohlin (2009) that “evidence-based software engineering may leverage on richer descriptions” through the integration and aggregation of these findings with evidence reported from similar situations.

7.7.1 Background to the I-E-S organization.

Figure 7-4 presents a high-level view of the six context facets applied to the general case. This section focuses mainly on the two outer context facets: organization and market. Although the figure presents a very high-level insight into the four inner facets, these will be addressed in considerably more detail when presenting the context of the specific projects investigated.
7.7.1.1 **Organization**

‘Integrated Enterprise Solutions’\(^{13}\) (I-E-S) is a multi-national software commercial off-the-shelf (COTS) product development organization. The case reported here describes activities performed within the research and development (R&D) division of this organization.

The R&D division comprises various teams dedicated to the development of different products. Employees work in both projectized and matrix structures depending upon their roles (PMI, 2008). Many software developers report directly to their project manager and teams remain together from project to project within the evolution of a particular product, thus growing both teamwork and domain skills. Quality assurance personnel may work on projects but report to an external manager, thus gaining the advantages of a common functional focus.

---

\(^{13}\) Integrated-Enterprise-Solutions (I-E-S) is a fictional name devised to protect the anonymity of the organization investigated.
The main R&D centres are located in the USA (multiple locations), UK, Belgium, Ireland, China and Australia, although virtual teams are also supplemented by individuals from sales, services and support offices in various locations throughout the globe.

### 7.7.1.2 Market

The R&D division is responsible for the development and delivery of various software products to the international manufacturing community. Licensed product installations are situated in over 5500 sites throughout the world. Products are targeted specifically at six industries: automotive, life sciences, configured goods, discrete products, consumer products and ‘food and beverage’. Other divisions within the organization include marketing, sales and services.

Having originally produced an enterprise resource planning (ERP) product, I-E-S has expanded its suite of product solutions to incorporate support for supply chain management, business intelligence, lean manufacturing, enterprise financials and other aspects of enterprise management for manufacturing organizations. I-E-S conduct two releases of its products per year: an interim release each February and a major product release each September. Products are classified as customizable core products or supplementary (‘bolt-on’) products. Also a number of third-party partner products may be used to further expand core-product functionality. Using configuration management practices and tools, the R&D development teams develop their products in code repositories that reflect ‘development branches’ and integrate their code into release-specific branches at pre-arranged dates prior to the February and September releases. (This enables the R&D quality group to conduct integrated system testing prior to product release).

### 7.7.2 Background to the application of Scrum within I-E-S.

R&D projects within I-E-S have been primarily conducted using predictive ‘waterfall’ type approaches. An internal toolset used to guide projects in the late nineties was characterized by a series of waterfall phases with formal reviews dictating project progression. However, even at that time, individual project managers had experimented with ASD techniques and this had been formally acknowledged by a project to extend that toolset to support such initiatives.
Notwithstanding that development, it should be stated that until relatively recently, the dominant approach performed by the organization was a predictive systems development lifecycle based method.

From 2007, variations of Scrum had been applied to some projects within I-E-S. Certain members engaged in one particular project (‘FIN-NN’), stand out as having more exposure to Scrum than any other group within the organization. The evolution of Scrum application within this project has occurred over two phases. An initial group of projects were performed by some team members using Scrum as interpreted from books and internal discussions. Unhappiness with their performance led to participation in a series of Scrum training classes. The processes were modified to reflect learning from these courses and since 2010; this particular sub-team have leveraged Scrum to perform their work within the FIN-NN project.

At the time of writing, there were a number of projects that leveraged the Scrum framework. Some project teams were collocated and others were distributed. Of these, one was used as a pilot for quality control purposes and it was deemed sufficient to confine the research scope to a detailed investigation of two projects.

(i) REF-ARCH: This project was a green-field initiative to develop reference architecture for internal developers within the organization. The team was distributed across the US, Belgium (Antwerp) and Ireland (Limerick). A member of REF-ARCH was used as a pilot to test the data collection procedures.

(ii) ‘FIN-NN’. This was a series of projects on a financial product. As stated above, this team had the most experience of using Scrum. The team configuration applied in this situation is termed ‘Scrum of Scrums’ (cf. section 4.3) and team members were distributed across Europe and Asia. Given the greater maturity of Scrum application within this team, it was desirable to include their activities within this case study. In order to determine the most appropriate work to be examined, a series of interviews was conducted with the manager responsible for this domain-area. He recommended that it would be most effective to
examine the activities performed by one particular sub-team of the overall FIN-NN team: ‘Fin-Limk’.

(iii) ‘QUAL-NN’: This project (or projects) addressed quality-related activities involving many of the products developed by I-E-S. This project is performed by a distributed team (the QUAL team). The notable point about this team is that it is configured in a structure referred to as ‘Totally-Integrated-Scrum’ (cf. section 4.3). This contrasts with the configuration used in ‘FIN-NN’. As a result, it was deemed desirable to investigate the activities of this particular project.

More recently, Scrum has been officially rolled out across the R&D division. A phased approach is being followed to implement the roll-out:

(i) Phase 1 (creation phase) took place from Jan-Mar 2012. Approvals were received and processes established. Although configuration of the Scrum framework is not an acceptable approach to the application of Scrum according to the Scrum guide (Sutherland and Schwaber, 2011), I-E-S decided that some framework tailoring should be performed to suit their situational needs.

(ii) Phase 2 is a pilot to monitor the performance of four global projects using the Scrum framework. Both the FIN-NN and QUAL-NN projects have been selected for this initiative and this is another reason for the identification of these projects as good candidates for the case study. The pilot process is managed by a Scrum process management team. The R&D vice-president and all directors have achieved Scrum Master certification and it is intended that all project and product managers are to achieve this accreditation. All employees in R&D are to receive three phases of training to ensure clear adoption of the tailored Scrum process: ‘Scrum Basics’; ‘Scrum in I-E-S (Process)’; ‘Scrum in I-E-S (Tools)’.
<table>
<thead>
<tr>
<th>Scrum Rollout Phase</th>
<th>Activities</th>
<th>Part of case study data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Establishment of processes, tools and training programs. Selection of Pilot projects</td>
<td>Use of processes, training and tools to enrich data collection and analysis</td>
</tr>
<tr>
<td>2</td>
<td>FIN-NN project</td>
<td>Scrum-of-Scrums GSD configuration. Main focus on work of one particular sub-team: Fin-Limk</td>
</tr>
<tr>
<td></td>
<td>QUAL-NN project</td>
<td>Fully-Integrated-Scrum GSD configuration.</td>
</tr>
<tr>
<td></td>
<td>Project 3</td>
<td>Not in case study</td>
</tr>
<tr>
<td></td>
<td>Project 4</td>
<td>Not in case study</td>
</tr>
</tbody>
</table>

Table 7-3: Formal Scrum adoption program in I-E-S

7.8 Summary

This chapter provided details on the design and execution of the selected research method (case study). Initial considerations such as fieldwork preparations are explained. This is followed by a detailed description of research design decisions. A ‘positivistic-like’ approach is taken to regulate the interpretive process. Each of the three stages of design setup, data collection and data analysis are described. Finally the chapter concludes with a description of the overall case context. This description will be further elaborated in the next two chapters as they present findings related to specific projects within I-E-S.
Chapter 8 - Distributed Scrum of Scrums Findings

8.1 Introduction

Figure 8-1 highlights the work packages performed in order to present a description of findings for empirical research within a project. These descriptions form a report of the empirical research findings, but it should be noted that another work package (2.3) from the overall WBS (Figure 1-7) relates specifically to case reporting. This final work package is presented in chapter 10 and it reports on findings from a combination of the projects within the case.

![Figure 8-1: Conduct Empirical Research WBS segment](image)

The goal of this chapter is to present a rich description of the empirical findings derived from research into the activities of a project (FIN-NN) performed using the Scrum framework by a distributed team using a ‘Distributed Scrum of Scrums’ configuration. Section 8.2 presents a detailed description of the project. The overall FIN-NN project team consists of four sub-teams.
This research is confined to an examination of the projects performed by one specific sub-team (Fin-Limk). This is followed by a detailed description of findings derived from the investigation. These findings are then used to refine the initial candidate conceptual LدScrum model that has been constructed from the work presented in chapters 2, 3, 4 and 5.

Sections 8.3, 8.4 and 8.5 present findings associated with NPD, ASD and GSD goals respectively. These findings reflect the goal rationale between core sprint practices and associated goals. Using the same structure applied to the presentation of candidate goal rationales in chapter 5, each of the sections is further decomposed into Scrum method fragments so that the goal rationale of each fragment may be reviewed independently. Note that although only three Scrum method fragments are proposed in chapter 5 (‘pre-sprint’, ‘sprint’ and ‘post-sprint’), it is necessary to analyse a fourth fragment (‘General considerations’) in order to ensure that goals that motivate all core sprint practices are considered. Each of these sections concludes with a table that presents an overview of the distribution of findings across the different goal practice associations. This table provides a high-level view of findings clusters and outliers that support the idiographic details presented in the section body. It was also used to verify the analysis process by comparing findings totals against the raw data to assure coverage accuracy.

Goal rationale analysis is followed by a review of the LSD values that emerged from the empirical findings. Section 8.6 contrasts the candidate value rationale for each LSD value against the refined value rationale deemed to have been present in the recollections and interpretations presented by the Fin-Limk team. The detailed findings of both the goal and value rationale are synthesised in Section 8.7. The two tables in this section present a refined goal rationale and value rationale that contrasts the proposed candidate associations with the empirical findings.

8.2 FIN-NN: Distributed Scrum of Scrum Context

Arising from the recommendations of (Petersen and Wohlin, 2009), section 7.7 presented a description of the I-E-S organization. They made the point that companies do not work with
standard methods but tend to tailor them to suit their needs. This view is a key tenet underpinning the concept of ‘method-in-action’ (Fitzgerald et al., 2002). They also noted that companies work in many different situations. As a result, they stated that in order for findings to be useful, it is vital to provide a comprehensive description of the study context. This section enriches the high-level I-E-S context by presenting a detailed description of the FIN-NN project. An initial overview of the FIN-NN project background outlines the evolution of the application of Scrum by this project team and the evolution of the team into a series of distributed Scrum of Scrums sub-teams. The four inner context facets described in section 7.7 are used to present detailed information on the activities of this particular team in relation to the overall FIN-NN project.

8.2.1 Background to the FIN-NN Project

The FIN-NN project is tasked with delivering functionality for the enterprise financials product. Initial development of the product was performed by a third party that was acquired by I-E-S. The original development team was subsumed into the I-E-S R&D organization and formed a core group in Belgium (Fin-Belgium). This group was supplemented by additional developers from different locations throughout the globe. As stated earlier, the standard ISD approach applied by the majority of R&D development projects within I-E-S used a waterfall structure and this was the method-in-action applied initially in the development of this product. A major constraint is that the product is built on a complex development environment that requires developers to conform to guidelines and design and coding approaches mandated by a proprietary framework. Fin-Belgium had first attempted to use Scrum in 2007 in an effort to support the decomposition of their work into small deliverables. This first initiative was not very successful due to limited training and pressures on the development team from customer needs.

“We found it a good idea to break the work into small deliverables. The first initiative failed. Our research on Scrum was just light... Easy to read but difficult to implement... a very difficult time for the team as it went to 10-15 customers and there was no stability.”

Domain-area Manager (Also Scrum-master of Scrum-masters)
The current team is composed of four sub-teams based in Antwerp (Belgium), Limerick (Ireland), Dudley (UK) and Shanghai (China). As stated above, this overall team of over forty developers emerged from the core group in Belgium. In 2010, the manager of the overall team went through formal Scrum training in order to better appreciate how to manage successful adoption of the method. Insights gained from this training led to the Antwerp team (Fin-Belgium) being trained in the method and this team applied the Scrum approach for the 2011-12 release timeframe. Initially, this team started with a pilot of three sprints. The pilot was very successful. Following this, a Limerick-based team (Fin-Limk) was added to the overall team in June 2011. Fin-Limk conducted three sprints adding some small features to the overall product. This was received very well and they received positive feedback on these early efforts. In September 2011, a Shanghai-based team (Fin-China) was formed and commenced development of various features for the overall product. Finally, in February 2012, a Scrum team based out of Dudley in the UK (Fin-UK) was formed. Each of the four teams work on independent features, integrating their code to a ‘product branch’ at regular intervals. The product branch is integrated to an overall I-E-S R&D release branch twice a year. One could characterize the structure of the overall product team as a ‘Distributed Scrum of Scrums’ which is considered a best practice and is recommended for distributed Scrum development by the Agile Alliance.

“Each team is separate. The only interaction that they had was doing sprint reviews and architect reviews. They are really separate teams working separately.”

Domain-area Manager (Also Scrum-master of Scrum-masters)

As per the Scrum framework, each team contains a product owner (PO), Scrum-Master (SM) and a set of developers whose primary skill-set is either a business analyst (BA), software engineer (SE) or quality assurance specialist (QA). Initially, product managers were assigned the roles of PO for each team. At the time of this decision, there were only three Scrum teams: Antwerp, Limerick and Shanghai. Two of the product managers were located in Antwerp and another based in Limerick. Due to product managers being tasked with many outward customer-facing responsibilities, they found it challenging to support the development team. The product manager based in Antwerp was also tasked with the PO role for the Shanghai team and difficulties occurred with this relationship due to the time zone difference. As a result of these difficulties, it
was determined more appropriate to select experienced business analysts to play the role of product owner on the different teams. Currently, the product owner of the Antwerp team is based in Dudley, UK. The Limerick, Shanghai and Dudley teams are collocated with their product owner.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PO</td>
<td>0.5 (located in UK)</td>
<td>1</td>
<td>1</td>
<td>0.5 (same PO as Antwerp)</td>
</tr>
<tr>
<td>Scrum-Master</td>
<td>QA</td>
<td>SE</td>
<td>QA</td>
<td>SE</td>
</tr>
<tr>
<td>Developer (BA)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Developer (SE)</td>
<td>4</td>
<td>3.5</td>
<td>3.5</td>
<td>3</td>
</tr>
<tr>
<td>Developer (QA)</td>
<td>1</td>
<td>1.5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 8-1: Configuration of I-E-S R&D financials project Scrum development team

Given that the team configuration was ‘Distributed Scrum of Scrums’, it was deemed sufficient to limit investigations to one of the four sub-teams: the Limerick based team (Fin-Limk). All the members of this sub-team were collocated. The product owner (PO) conducted his work in a cubicle embedded within the developers. At full strength, the team contained a PO and seven developers. The developers represented core competencies in business analysis (BA), software engineering (SE) and quality assurance (QA). One SE and QA could only dedicate 50% of their time to team activities as they had a wider remit within the R&D organization as a technical architect and QA manager respectively.
Figure 8-2: Fin-Link team in dedicated Scrum meeting room using JIRA

All team members worked in close proximity (low-partition cubicles) and had access to a room which contained the Scrum board, projection equipment and dedicated video-conferencing facilities for events such as post-sprint reviews. The following subsections leverage checklists associated with the four inner context facets proposed by Petersen and Wohlin (2009) in order to present details on the context of this team’s work environment.

8.2.2 Product

The financials product provides an integrated solution that addresses reporting and financial management requirements of global manufacturing organizations. Features such as automated accounts consolidation and workflow management enable efficient financial management. Other key benefits provided by this system include management information visibility, multi-currency management, regulatory compliance, tax and audit management. The product comprises of a number of modules and is customizable. It may be integrated with a number of add-on modules that provide additional functionality such as tracking to support compliance with Sarbanes-Oxley (SOX) requirements. Additional modules are available from third-party organizations that
partner with I-E-S. It is a mature product that has gone through many years of releases and is in use in many countries world-wide. Development of the product is under stringent quality requirements that are assessed at different levels of development both within the core development team and a wider I-E-S system integration and release quality assurance team. Core development of this product is undertaken using Java and C sharp. Ongoing development leverages a core framework that forms the foundation for overall product design and development.

8.2.3 Processes

As stated in the outset of this chapter, the overall R&D organization of I-E-S conducts two release activities per year. Both the interim release in February and the major product release in September incorporate many different features and products. One such product is the financial product described above.

Requirements in this product are formed from the work performed by a product management team. A roadmap of features is established and augmented on a regular basis. The Scrum framework is the development method used by all sub-teams within the FIN-NN team. The features roadmap forms the ‘product backlog’ that guides the development teams in determining the work to be performed during their development sprints. In order to enable effective communication between teams, sprints are staggered so that two of the four sub-teams commence their respective sprints one week before the other two teams. The organization is ISO-certified and all R&D projects are subject to rigorous documentation of agreed artefacts that guide development and are subject to ISO audits. Each of the four FIN-NN Scrum sub-teams is responsible for a major sub-domain of the financials product domain. For example, one particular Scrum team is responsible for all aspects of asset management. Configuration management of the financials product stipulates that a single code repository forms the baseline branch which is ultimately released as a generally available (GA) product. This branch also forms the basis for subsequent maintenance of that GA product. Each FIN-NN Scrum sub-team produces working features for the domains that are their responsibility within a code repository dedicated to their work exclusively. At periodic intervals (every third or fourth sprint), these
dedicated code repositories are merged into the financials baseline repository. A number of weeks prior to the February and September release dates, the release baseline code is frozen in order to undergo system testing in support of the release process.

8.2.4 **Practices, Tools, Techniques**

The team worked in three-week sprints. Each sprint was launched by a sprint planning meeting on the Monday morning of the first week. In the week prior to the beginning of the sprint, the Fin-Limk PO worked with the developer whose core competency is as a business analyst. These two individuals reviewed the product backlog to produce a set of epics which provide greater detail on certain high-priority requirements. The sprint planning activities determined how the team would address these epics in the sprint. In order to do this, the Fin-Limk team assembled in the dedicated Scrum meeting room. Planning activities were initially performed using post-it notes and the Scrum board but this practice has evolved to leverage software tools projected to a screen. The planning process decomposed product backlog requirements into user stories. Having projected epic requirements on the screen, the Scrum-Master took suggestions from team members on how best to construct user-stories that would result in the features needed to address the requirements. Once all the user stories were completed, the team would play ‘planning poker’ by using storypoint cards to propose independent estimates for each user story. This technique promoted debate and eventual consensus on the number of story points that should be allocated to each user story. This information was then used to determine what work can be completed by the team in the sprint based upon their development capacity. The product owner assisted in this selection process by prioritizing stories as necessary. Once agreement had been reached on all the work to be completed, the selected stories were entered as work tickets in Atlassian JIRA. This product identifies and records the sprint backlog. This enables progress tracking and provides records for traceability and auditing purposes following sprint completion. This initial planning was normally completed by lunchtime on the first day of the sprint. The remainder of that day was used to elaborate on the detailed planning of each story. Stories were decomposed into tasks and time estimates were assigned to support progress measurement within the JIRA tool. Acceptance criteria were then established for each of the user stories. These criteria were built with input from all of the team and formed part of the ‘definition of done’ for the sprint.
This work could involve discussions on the design and quality assurance needs of different features.

Development activities were performed from days two to fourteen of the sprint. These activities were supported by the close-knit development environment that promoted collaboration and accessibility. There were also many empty meeting rooms available. Further control of team progress was afforded by a daily Scrum that was held at 12:30 each day.

Various meetings were held among distributed developers during the sprint. The distributed Scrum-of-Scrums team conducted meetings every Tuesday and Thursday in order to coordinate and control overall team activities. These meetings were attended by the Scrum-master from each remote team and the domain manager who acted as the Scrum-Master of Scrum-masters. This meeting used the JIRA tool to generate a burn-down chart that reflected the performance of each team within their respective sprints. Another important development meeting was a weekly product owner meeting. Individual team members also participated in R&D development meetings involving participants from both Scrum and non-Scrum projects. An example of such a meeting was the weekly technical architect’s meeting.

Each sprint concluded on the Friday of the third week. A review of the working software was presented to the wider distributed team and external parties such as support, services and senior product and development management. Given that there were four sub-teams in the wider FIN-NN team, there was not enough time to effectively conduct four post-sprint sessions. Therefore, sprints were staggered so that only two teams ever completed a sprint on the same week. All teams were present for all sprint reviews. In the case of Fin-Limk, the Scrum-master presented a sprint summary outlining completed features and any planned features that may not have been produced. Then another member of the team would demonstrate the completed features and take questions from attendees. Finally the meeting concluded with an architect’s review. The final sprint activity performed was the team retrospective. Each sub-team conducted their own sprint retrospective and the only attendees were the sub-team and their particular product owner.
As stated above, I-E-S is ISO-9000 certified. R&D operations are subject to audit and a key element scrutinized is individual development projects. Documentation such as requirements specifications, design documentation etc are stored in a central repository. In the context of the FIN-NN teams, there were repository pages associated with each team. These pages contained documentation on sprint details including initial epic details, sprint summaries, sprint burndown charts and retrospective notes and decisions.

8.2.5 People

Although numbers may vary from time to time, there are approximately forty developers spread across the distributed FIN-NN team. The Fin-Limk team is quite an experienced team. Just one developer has less than twelve years software development experience and most of the team have extensive experience in the domain. All team members have strong technical training backgrounds and have been formally trained in the Scrum framework and its application within the FIN-NN project. The product owner is a member of the product management team. Two of the software engineering developers are considered technical architects within the wider R&D organization and the quality assurance developer divides her responsibilities within the Scrum team with her duties as a manager of the quality assurance group across R&D. Another very experienced quality assurance engineer is temporarily unavailable to the team. The strong experience levels within this team led to an issue that resources needed to be involved in additional activities such as QA management or R&D technical design reviews. This has been addressed by the addition of two software engineers.
8.3 Findings associated with New Product Development Characteristics

The investigation into the presence or absence of characteristics associated with successful new product development (NPD) was managed through a central hypothesis:

“A distributed team that uses Scrum will exhibit all 6 characteristics of a successful new product development team (Built-in Instability, Self-organization, Subtle control etc.)”

The approach taken analysed interview data in order to reveal instances where activities were performed in order to achieve a goal that would reflect an NPD characteristic. A key aspect of the analysis was the identification of the method fragment that contained the practices performed to achieve the goal. In this way, comparisons could be made between the method fragment portions of the candidate LDScrum model and the empirical data in order to propose refinements to that portion of the model. As stated in chapters 6 and 7, the intention of declaring a hypothesis was not for falsification purposes - it was to leverage the hypothetico-deductive approach in order to structure the analysis of rich descriptions so that said analysis could inform the refined LDScrum model. Quotations from interviews are presented here to support the findings. The role of the source of each quotation is provided. Please note that all quotations came from the Fin-Limk sub-team.

Chapter 5 outlined the LDScrum model and presented the candidate relationships between LSD values, NPD, ASD and GSD goals and Scrum practices. In order to simplify and clarify the management of this complex set of relationships, the LDScrum relationships are presented at the method fragment level: pre-sprint, sprint and post-sprint sets of practices.

This model was constructed in part from the work presented in chapters 3 and 4 which produced subsets of these relationships, linking NPD, ASD and GSD goals to Scrum practices (the ‘DScrum’ model). It is the relationship between NPD goals and Scrum practices that is the concern of this particular section. This relationship is summarized in table 3-11. Analysis of the empirical data synthesized the practices into the three aforementioned method fragments and also
grouped three outstanding core Scrum practices into a fragment called ‘general considerations’. The findings for each of these fragments are contrasted with the proposed candidate relationships in the following sections. To support this comparison, each section commences with a table that contains both the candidate goal rationale and refined (empirical) goal rationale for the method fragment addressed by the section. Each candidate practice-goal association is denoted by a tick mark (✓). Each empirical practice-goal association is denoted by the letter ‘E’ (Empirical finding). Instances where a practice-goal association is present in both candidate and empirical results are shown by the presence of both the tick and the letter ‘E’ in the cell.

### 8.3.1 General considerations

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>NPD Goals</th>
<th>New Product Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-1</td>
<td>Scrum Team Configuration</td>
<td>E</td>
</tr>
<tr>
<td>CS-2</td>
<td>Initial planning (and ongoing) - create or update product backlog</td>
<td>✓</td>
</tr>
<tr>
<td>CS-12</td>
<td>Definition of Done</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 8-2: Candidate NPD goals pursued by the ‘General Considerations’ fragment (from table 3-11)

Initial analysis proposed that all of the NPD goals were pursued by practices within the general considerations fragment. However, these expectations were not the experiences or interpretations recounted by members of the Fin-Limk team.

Embracing the freedom to address challenging targets was observed by just one individual on the team. In the main, most individuals did not acknowledge the need for built-in instability (NPD-1) to be a motivator for the team configuration. This particular goal was much more evident in their observations on the reasons that they performed particular pre-sprint activities. This will be described in greater detail in the next section.

Just two of the team members expressed a view in relation to the impact of self-organizing project teams (NPD-2) on general considerations practices. Of these two, one team member’s
observations contained an element of apprehension on the effectiveness of Scrum team configurations to meet this goal:

“I accept that you are giving more autonomy to the teams but this depends upon the team’s ability. The core knowledge still resides in XX\textsuperscript{14}. We want to have sharing between teams also”

Developer (SE)

Only two respondents noted general considerations practices to be motivated by a goal to achieve collective responsibility (NPD-3). Of these two, there was a stark contrast in their reflections on the effectiveness of general considerations practices to meet the goal. One team member felt that the Scrum team configuration practice (CS-1) achieved a goal of collective responsibility:

“It was clear there was not specific roles in the Sprint team...everybody still had their speciality. I was hired as a BA. Everybody battled with that at the beginning but we have gotten over it.”

Developer (BA)

On the other hand, the only other observation on NPD-3 described how managing the product backlog (CS-2) was ineffective at achieving the goal:

“The PO and BA are ... involved in the pre-sprint. They have pushed to get the team involved a week before. The team have pushed back ... we don’t want to take away from the current sprint”

Developer (SE)

The only other goal or NPD characteristic that was deemed to motivate general considerations practices was subtle control (NPD-5). This was also only noted by two respondents (SE and product owner). In general, there was little attention paid by the overall team to the NPD goals when considering why they performed general considerations practices. However, this ambivalence to NPD goals is reversed in their observations on the influence of NPD goals on the practices of the pre-sprint method fragment.

\textsuperscript{14} XX: Team name (location) is omitted to preserve anonymity
8.3.2  Pre-sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Scrum Practices</th>
<th>NPD Goals</th>
<th>New Product Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-3</td>
<td>Sprint planning - create release backlog</td>
<td></td>
<td>NPD-1 NPD-2 NPD-3 NPD-4 NPD-5 NPD-6</td>
</tr>
<tr>
<td>CS-4</td>
<td>Sprint planning - create sprint backlog</td>
<td></td>
<td>E E√ E E E√</td>
</tr>
<tr>
<td>CS-5</td>
<td>Sprint planning - define sprint goal</td>
<td></td>
<td>√</td>
</tr>
</tbody>
</table>

Table 8-3: Candidate NPD goals pursued by practices in ‘Pre-Sprint’ method fragment (from table 3-11)

Discussions around the performance of sprint planning revealed a much stronger awareness of the influence of NPD goals on these practices. Built-in Instability (NPD-1) was observed by all team members to be a motivator for creating the sprint backlog (CS-4). Many of the developers commented on the challenging nature of requirements and how the freedom given to them to select appropriate work items resulted in a tension that was managed through practices such as planning poker with user stories:

“Everybody has their deck and they pick a number and put it face down on the table. Somebody says "are we ready" and we turn it over. The rule that we apply is the person with the highest justifies theirs and the person with the lowest justifies theirs. And sometimes we vote or sometimes people just concede. But we do find merit in it in that complexities that somebody might see may have not been anticipated by others”

Developer (SE)

“When we all use the planning poker cards and we could all agree or there can be one person has a 2 and somebody else has a 13...we need to have a debate on this big difference.”

Developer (SE)

Further evidence of the establishment of challenging targets and releasing control to the team are seen through the eyes of the product owner and his relationship with the team:

“I usually oversubscribe so that folks don’t run out of work. They define the user stories and then I get to set the priority of the user stories.”

Product Owner
The independence and freedom granted to the team in this development process is evident from the acceptance of developer autonomy as described by the business analyst:

“Yes - it could happen (that the PO might question an estimate). But he would have to concede.....The software engineer would say what it is and generally - you can’t argue with them. They know the code.”

Developer (BA)

Such developer autonomy combined with knowledge sharing and extension of goals reflects a goal to be a self-organizing team (NPD-2). This goal was evident in comments by many developers around sprint backlog planning (CS-4). The autonomy to determine how best to tackle planning tasks is seen from observations on how the developers modified the process to suit their situation:

“Our planning was not deep enough. We have added ... an Acceptance Criteria meeting. All the participants for a task delve into a story. Planning is done when you have all your acceptance criteria .”

Developer (SE)

Process modifications not only reflect autonomy but also an awareness to embrace continuous learning. Another experience reflects this awareness of the need to accept that experience improves the planning process as the team matures:

“We tried to make them only one story point but then our stories were overlapping too much. This caused everything to come up to test in the last week which did not work.”

Developer (SE)

This is further endorsed by the observations of another developer on the need to accept that continuous learning and process evolution has been most effective in their experience:

“If something has too high story points we consider it too big to do in 3 weeks...you have to do it the hard way and make the mistake and find it out during the sprint and take it on to the next planning time.”

Developer (SE)
The third aspect of a self-organizing team (NPD-2) is knowledge sharing. One developer recollected how this particular aspect of self-organization motivated the modification of their approach to user story acceptance criteria creation in the sprint backlog planning practice (CS-4).

“Before we used to only do it (acceptance criteria meeting) with whoever was working on the story but now we bring in everybody because people have an input and it’s valuable to have everybody’s say.”

Developer (SE)

Although not highlighted by all team members when discussing sprint planning, the characteristic of overlapping development phases or collective responsibility (NPD-3) appears to be implicit in much of the general approach to their work. One instance of sprint backlog planning that highlights this is the recollection of a software engineer about the effectiveness of having different roles collaborating on the estimation process:

“Sometimes I would not have thought of how much testing might be needed and QA\(^{15}\) would speak up and highlight the need for more testing”

Developer (SE)

Respondents did not make many references to learning across functions (NPD-4). However, there are times when either individuals or the entire team needed to grow their domain or technical knowledge. In such cases, sprint backlog planning resulted in specific tasks being identified and planned to manage that learning. The concept of subtle control (NPD-5) is referred to in the context of interactions between the team and product owner. The product owner allowed the team to determine how much work is involved in completing certain requirements but tempered this non-invasive management with the right to re-prioritize the requirements in order to get a commitment on the completion of particular pieces of work.

“We go back to the PO...explain that we won’t get all the work done after breaking down the requirement.. The PO accepts this ... he prioritizes the user stories to see what has to be done for this sprint”

Developer (SE)

\(^{15}\) QA: Quality Assurance developer name omitted to preserve anonymity
8.3.3  
**Sprint**

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>NPD Practices</th>
<th>New Product Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td>√ E √ E E E √ E</td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td>√ E √ √ E</td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td>√</td>
</tr>
</tbody>
</table>

Table 8-4: Candidate NPD goals pursued by practices in ‘Sprint’ fragment (adapted from table 3-11)

To support this investigation of NPD goals that motivate certain practices, an additional aspect of the research is to interpret whether or not the execution of such practices is seen to achieve the goal. Although one developer reflected that self-organizing teams (NPD-2) was an influence on the performance of general sprint activities (such as daily Scrum and other meetings), he felt that the goal was not always achieved. For certain requirements, he noted that general Scrum activities do not lead to the achievement of cross-fertilization! The SE stated that there is not as much learning or sharing going on within the team as one would assume. In such situations he felt that the BA, PO or QA roles would not care how he developed something.

“Sometimes you just concentrate on an epic you have been given ... There is not that much transparency - there is a bit...Maybe everybody is in their own little waterfall during their task”

Developer (SE)

However, it should be noted that despite the above observation, overlapping development phases (NPD-3) is the goal most cited by the team in the context of Sprint practices. It is considered a motivator that promotes collective responsibility and helps reduce delays. The business analyst recollected tackling design meetings with multiple roles in order to share knowledge and also the product owner and multiple software engineers commented on the effectiveness of overcoming a shortage of QA resources by having software engineers perform testing activities.
"We generally have a couple of people - a BA, a QA and an SE or 2 in the room.. we will be cracking
open the s/ware and ... battling over and back ...”

Developer (BA)

“In other previous sprints, we have not had enough QA resources so some of the developers have moved
in and done some of the testing work... We try to work on any of the tasks that are available ”

Developer (SE)

Individual and team learning (NPD-4) was pursued in general sprint activities. The daily Scrum
was highlighted as a key practice that informed about areas of focus for different team members
and enabled an individual to seek learning from the correct parties, when necessary.

Subtle control (NPD-5) also promoted certain Sprint practices. The product owner explained that
he must relinquish freedom to the team and allow them to perform the work as they see fit.

“That’s the beauty of Scrum - you can’t delve in . That is what we used to do in Waterfall. If BA\textsuperscript{16} comes
back with something that has to change, then it has to change”

Product Owner

However, in cases where he observed that the work was losing focus, he was in a position to
curtail any scope creep. He ensured that although the team had freedom to determine how to do
the work, they needed to remain within the boundaries established at the start of the sprint.

“I felt we were doing too much functionality for one or two customers. You have to balance business
needs with the cost of the work.”

Product Owner

\textsuperscript{16} BA - business analyst name is removed to preserve anonymity
8.3.4  Post-Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>NPD Goals</th>
<th>New Product Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-10 Post-Sprint - review meeting</td>
<td>E √</td>
<td>E E E √</td>
</tr>
<tr>
<td>CS-11 Post-Sprint - team retrospective</td>
<td>E √</td>
<td>√ √</td>
</tr>
</tbody>
</table>

Table 8-5: Candidate NPD goals pursued by practices in ‘Post Sprint’ method fragment (from table 3-11)

Each of the candidate NPD goals associated with post-sprint activities (table 8-5) is present in the empirical research. One additional goal also emerged from this analysis: Subtle Control (NPD-5). NPD-1 was found to be a goal for post-sprint practices because these practices helped persuade management to release control to the team.

“In the past (senior management) used to ask what are we doing here, there... Now they only ask what we will deliver. This seems to be down to the demo and retrospective.”

Developer (SE)

The team moved the time and day of their retrospective (CS-11) a number of times in order to best meet their own needs. Such autonomy to situate a key practice in the most effective way is an example of self-organization (NPD-2). Findings related to the other three goals are not very convincing. Some developers indicated that demos (CS-10) were motivated from a desire to conduct multi-level learning between individuals, teams and even to the wider organization. The product owner considered learning to be a low-level motivator for this practice. However, where all respondents are united is in their belief that learning is not being achieved through this practice.

“Sometimes I find it a complete waste of time. We try to give an overview of the demo and what we are going to do. XX just jump straight in to going through screens. You have no big picture. ....”

Developer (SE)

However, the review demonstration (CS-10) was seen to pursue the goal of subtle control (NPD-5) from two perspectives:

17 XX: Team name (location) omitted to preserve anonymity
Presenting the work achieved is seen as a tool to protect the team from invasive management techniques as it displays the team’s competence.

Because this practice is perceived to be a management review, it acts as a form of subtle control by forming the boundary of the space in which the team is empowered to work.

“At the minimum it needs to be working... you have put everything into finishing for the Friday review and for it to go wrong on the day is not good”

Developer (QA)

More respondents recollected organizational learning (NPD-6) as a motivator for post-sprint practices than any of the other NPD goals. The main point raised by the team was that the presence of senior management and members of external groups at the demonstrations enabled the team to share information and receive valuable feedback. Although most observations in relation to this goal referred to the post-sprint practice of sprint review (CS-10), one developer considered how the sprint retrospective (CS-11) might be motivated by this goal. They accepted that the motivation was very low currently and that the goal was not being achieved. However, it was an interesting and useful observation that could lead to additional value being leveraged from that particular post-sprint practice:

“Perhaps we should be sharing things that we find very good (externally). We have not done that to date... Perhaps there could be some cross-fertilization of what comes out from those meetings.”

Developer (SE)

### 8.3.5 Summary of NPD Findings

The characteristics of a successful new product development team are found to be present in the distributed Scrum team. These characteristics are seen as objectives that motivate various practices performed by the team and in many cases appear to be realized.

Table 8-6 summarizes the findings from this work. Although it does not tally with the initial candidate set of goal-practice mappings, the table does reflect that the overall landscape of practices is related to all of the NPD characteristics. The columns in the table represent the
method fragments or groups of practices. The rows represent each NPD characteristic. Each cell in the table represents a method fragment and NPD characteristic intersection. It presents the number of findings where a practice from the method fragments was seen to be motivated by a desire to achieve the NPD characteristic. Some responses indicate times when a goal may be pursued by a particular practice but not achieved. These instances are denoted by an asterisk. Other responses have prompted the issue that the performance of a practice may only be motivated in part by a particular goal or even in some cases; the performance of the practice may contradict the goal. Part motivation is indicated by a single exclamation mark whereas direct contradictions are noted by a double exclamation mark. In this section, there were no instances identified where the performance of a particular practice contradicted an NPD goal.

It must be stressed that this is not a positivistic study of a phenomenon. The use of a ‘hypothesis’ is merely as a vehicle that permits more meaningful analysis to be applied in a consistent manner. The position taken is that the summary landscape of NPD characteristics is too shallow a measure and that the more important knowledge to emerge from this exercise is the meaningful rich descriptions and observations made by various team member’s in relation to their work practices.
<table>
<thead>
<tr>
<th>GENERAL CODE Description</th>
<th>DScrum Goal</th>
<th>Sub-characteristics</th>
<th>Practices</th>
<th>Gen</th>
<th>Pre-sprint Findings</th>
<th>Sprint Findings</th>
<th>Post-sprint Findings</th>
<th>NPD FRAMEWORK EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPD1 - Built-in Instability</td>
<td>g1</td>
<td>Challenging Targets</td>
<td>CS1; CS4; CS10; CS11</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>Setting challenging targets and releasing control of how these targets should be met introduces a tension and instability that empowers teams, promoting creativity and innovation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Release of control to team</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPD2 - Self Organizing Project Teams</td>
<td>g2</td>
<td>Autonomy</td>
<td>CS1; CS4; CS7, CS11</td>
<td>2*</td>
<td>6*</td>
<td>1*</td>
<td>1</td>
<td>Autonomy empowers the team to manage their own situation without external interference. Self-transcendence is evident when teams practice continuous improvement via the setting and extending of goals in pursuit of the higher-order challenges set down by senior management. Cross-fertilization occurs as teams share knowledge.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Self-transcendence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cross-fertilization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPD3 - Overlapping Development Phases</td>
<td>g3</td>
<td>Learning across functions</td>
<td>CS1; CS2; CS4; CS5; CS7, CS8</td>
<td>2*</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>promote collective responsibility and the reduction of potential delays. Such delays may be due to bottlenecks and misunderstandings associated with the ‘over the wall hand-off approach associated with division of labour using separate specialized development phases.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learning at different levels (individual, group, corp.)</td>
<td>CS4; CS7; CS10</td>
<td>0</td>
<td>2</td>
<td>3*</td>
<td>3*</td>
<td></td>
</tr>
<tr>
<td>NPD4 - Multi-level learning</td>
<td>g4</td>
<td>Freedom within boundaries</td>
<td>CS1; CS4; CS7; CS10</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>encompasses the notion of learning being promoted at multiple levels (individual, group, corporation) and that individuals seek learning across functions (such as ISD and marketing).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tolerant non-invasive management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPD5 - Subtle Control</td>
<td>g5</td>
<td>Sharing via formal artefacts</td>
<td>CS7; CS8; CS10; CS12</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6*</td>
<td>Subtle control refers to the management of self-organizing teams by permitting freedom within boundaries that are marshalled through non-invasive management techniques that are tolerant and reflective of the uncertain non-linear nature of new product development.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Judicious assignment of resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Standardization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8-6: Overview of field results related to the presence of NPD goals

* indicates issues with goal achievement; ! indicates partial goal motivation; !! indicates goal contradiction
8.4 Findings associated with Agile Software Development Principles

The examination of the relationship between ASD principles and Scrum practices uses a similar structure and approach as that presented in the previous section (section 8.3). A central hypothesis is used as a device to frame the analysis. The exploration leverages interview data and other project documentation to collect reflections of team member’s on their performance of various Scrum practices. Individual observations are examined to determine if they are motivated by various ASD principles. An interpretation of the collective perception of the team about the relationship between ASD principles and Scrum method fragments is presented in tabular format. This high-level view acts as a lens so that the researcher may devise interpretations on the broader social reality at play within the distributed Scrum team. The ‘hypothesis’ (or premise) that is used to position the analysis is as follows:

“A distributed team that uses Scrum will achieve all 12 goals associated with the principles of ASD”

Clearly one would expect that the foremost ASD method would achieve the goals associated with ASD principles. However, as stated in chapter 3, Conboy and Fitzgerald (2004) noted that ASD methods do not always follow all of these principles. In any case, the general ‘hypothesis’ has been formulated as a general premise to support analysis on the question of which practices are motivated by which goals. Candidate relationships identified between ASD principles and Scrum practices (table 3-11) are contrasted with empirical findings associated with each of the four method fragments under investigation: general considerations, pre-sprint, sprint and post-sprint. In this way, a rich meaningful description of the situation emerges from an interpretation of the directed reflections of team members. This description may then contribute to the refinement of the overall candidate LDScrum model in order to inform future complementary research efforts into this phenomenon. As in the earlier sections, each section presents a findings table that shows both the candidate and refined goal rationales for the practices of that fragment. Conventions used in the earlier section apply here also. A tick mark denotes a candidate goal-practice association and an ‘E’ reflects an association uncovered in the empirical findings.
8.4.1 General Considerations

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum Team Configuration</td>
<td></td>
<td>E</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial planning (and ongoing) - create or update product backlog</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition of Done</td>
<td></td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8-7: Candidate ASD goals pursued by practices in ‘General considerations’ fragment (adapted from table 3-11)

The proposed candidate set of relationships are more extensive than the empirical findings. In the candidate table above, eight of the ASD principles are considered to be motivational goals that drive the pursuit of practices that form the ‘general considerations’ method fragment. However only five principles feature in the empirical findings.

Customer satisfaction (A-1) is considered a motivating goal by three respondents. The product owner identifies the general iterative nature of the framework as a mechanism that enables customer escalations to be addressed rapidly and working software to be delivered promptly.

“We have delivered early in numerous occasions by dealing with escalations.... in the case of escalations we can do work that is patched to the customer very quickly.”

Product Owner

Another developer contrasted work using Scrum to prior projects that employed a waterfall approach. Their impression of the general Scrum framework was that it appeared to be a more effective approach as all committed work was completed to meet the customer’s needs. In their experience, previous projects tended to conclude with intensive testing and a compromise agreement to release with deferred requirements. The developer appeared to have a distinct feeling that something worthwhile had been completed because the outcome was relevant to the customer:

“We delivered it...nothing got pushed to deferred requirements... everything that we committed to. There actually is a demo for that customer this afternoon - the actual customer rather than somebody internal”

Developer (QA)
Team structures and physical positioning was deemed to have been motivated by a need to convey information effectively (A-6). As the distributed team is configured as a ‘Scrum of Scrums’, all of the developers in the Fin-Link team are collocated.

“We are physically sitting beside one another. We even changed that a bit. We used to break into sub-teams. But we said we should all sit together to ensure that QA has visibility to what is going on”

Developer (QA)

Although not identified in the candidate set of goals, using working software as the primary measure of progress (A-7) was discussed by three of the respondents as a goal pursued by Definition of Done (CS-12). Two respondents felt that the concept of ‘working software’ was unclear and one respondent decried the lack of metrics that could be used to pursue such a goal. However, in contrast to this, one software engineer indicated that the goal led to careful consideration of what constituted a successful completion of their work i.e. working software.

“We are very careful about our definition of done. We will not demo something that is not rigorously tested - we will simply say we did not achieve it”

Developer (SE)

8.4.2 Pre-Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Scrum Practices</th>
<th>ASD Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>√</strong></td>
</tr>
<tr>
<td>CS-4</td>
<td>Sprint planning - create sprint backlog</td>
<td>E <strong>√</strong></td>
</tr>
<tr>
<td>CS-5</td>
<td>Sprint planning - define sprint goal</td>
<td></td>
</tr>
</tbody>
</table>

Table 8-8: Candidate ASD goals pursued by practices in ‘Pre-Sprint’ method fragment (from table 3-11)

The number of ASD goals pursued by pre-sprint practices was found to exceed the amount proposed by the candidate set shown in Table 8-8 above. Nine\(^\text{19}\) of the twelve goals were

\(^{19}\)Only 8 goals are shown in the table. An instance of ‘update product backlog’ was deemed to pursue A-4 but was interpreted as a pre-sprint planning practice in the context presented.
identified by one or more respondents. Sprint planning was seen to be strongly influenced by the goal that customer needs should be a top priority (A-1). Many of the developers either directly mentioned customer needs as being a key criterion to work prioritization or deferred to the product owner as the prioritization agent acting on behalf of the customer. The business analyst’s reflection on this area captured the sentiments expressed by many developers:

“We tried to maintain our customer focus a lot by using our PO. We would ... determine an approach and solution, the PO comments would be key to whether it was acceptable or not.”
Developer (BA)

Welcoming changing requirements (A-2) emerged very clearly as a strong influence on pre-sprint practices. Many of the developers and the PO expressed a very clear awareness and willingness to manage changing requirements. The following comment appears very assured and is quite representative of the general views expressed by many other team members:

“We 100% accept that they can come in with anything for the start of the next 3 weeks...once your sprint is over, done and dusted, you’ve demonstrated it and you’re onto the next thing... - a whole new day.”
Developer (BA)

A mixture of views was expressed in relation to the influence of building teams around motivated individuals (A-5). Certain respondents considered sprint planning (cs-4) to be motivational whereas one respondent questioned the relevance of story points to determine team capabilities.

“Doing similar work all the time you can associate a number with work without relying on time...we have been doing distinct and different chunks of work. I’m not sure how relevant these story points are”
Developer (SE)

As stated earlier, the group had a very clear perception and awareness of their willingness to embrace change and how sprint planning was used to achieve that goal. This clarity of purpose was also evident in relation to the goal of achieving sustainable development (A-8).

“We are getting there (achieving sustainable development). The pre-sprint - we don’t have very big stories any more. Maybe a 5 point story and we have a good idea of what you can pick off.”
Developer (QA)
The care taken in sprint planning is considered vital to the achievement of sustainable development (A-8) and this goal seems to be a major consideration of team members:

“Sprint planning...is the key meeting...the quality of the work that goes on in that...if that is good then the rest of the time is relatively easier. If there is a lot of fudge...you might end up with three weeks of attrition.”

Developer (BA)

As a consequence of issues within Sprints, user stories were tightened up to deal with the standard situation initially, followed by separate stories for exceptional cases. Thus, the pursuit of simplicity (A-10) motivated how aspects of sprint planning were performed.

“We thought it was not too bad but when we got into it, it was a way bigger than we thought. Every sprint after that, we broke out stories so that we did anything for ‘normal’ first then ‘staged’ second.”

Developer (QA)

### 8.4.3 Sprint

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td>E</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td>E</td>
<td>√</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>√</td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td></td>
<td></td>
<td>E</td>
<td>√</td>
<td>√</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>√</td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph</td>
<td></td>
<td></td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
</tr>
</tbody>
</table>

Table 8-9: Candidate ASD goals pursued by practices in ‘Sprint’ method fragment (from table 3-11)

Given that the sprint is the heart of the Scrum set of practices, it is perhaps understandable that many of the candidate ASD goals would be considered influential to practices within this method fragment. The empirical findings tallied with this expectation.

The pursuit of technical excellence (A-9) emerged as a goal that was influenced by the sprint activities of many team members. Experiences in relation to this goal were mixed. Certain team members expressed the view that within the sub-team, this goal was being pursued and achieved through regular meetings such as the daily Scrum and other general sprint meetings (CS-7).
“Reading the signs from them that something is difficult... trying to get them to be more open as to what those technical issues are... on the daily Scrum when they talk about yesterday's obstacles.”

Developer (BA)

However, other team members felt that pursuit of the goal of simplicity (A-10) contradicted the goal of technical excellence, compelling them to settle for non-optimal design solutions.

“We try to achieve that (Keeping things simple). I don’t like it. I have to ignore things that I see that I can improve.. It’s a tunnel view. It should be at least called out as an issue”

Developer (SE)

Certain comments were made in relation to the pursuit of technical excellence across the distributed team. Two senior team members explained that the goal was definitely being pursued by cross-team meetings but they did not feel that the goal was being fully achieved.

“We try to have a weekly meeting with a design rep from each team so that they can feed up on a weekly basis.. People just don’t highlight stuff. You discover some weeks later that things have happened that should have come up at these meetings.”

Developer (SE)

Although striving to preserve simplicity (A-10) was identified by some as contradicting the pursuit of technical excellence (A-9), it was still seen by many to be a goal that is pursued and achieved within general sprint practices. The following excerpts from members of the team with different core competencies highlight the prevalence of the pursuit of simplicity within their activities:

“we always say to keep it to the simplest solution possible. People would tend to say “what about if they decide to bring this in” and somebody would say “don’t think about that - it might never happen” ”

Developer (SE)

“The team can overcomplicate things and do more than is needed. That is why you need to have a business view so that you can balance the customer’s requirements against the cost of doing something”

Product Owner

“In the beginning developers are likely to say that "ah when I’m in here I will make the change as I have the code...". Then the QA would say "no-you’re only supposed to do this work". ”

Developer (QA)
Regular collaboration between business people and developers (A-4) was highlighted by many as an achievement of sprint practices. A consistent understanding on the role of the product owner emerged from discussions on this point. He showed a very clear awareness of his role as a representative of customers.

“I attend the sprint meetings most days...if there is a debate in the team, I make the decision...They deliver to me ... My responsibility is to deliver to the wider team. How far do we go into something?”

Product Owner

The manner in which the PO performed this role in conjunction with the BA was well accepted and consistently referred to by many different developers.

“Daily Scrum meeting. PO does not give status but answers questions. PO & BA meet weekly with our customer outside of the Scrum meeting. We would also communicate a lot outside of the Scrum meeting.”

Developer (SE)

8.4.4 Post-Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Scrum Practices</th>
<th>ASD Principles</th>
<th>ASD Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-11</td>
<td>Post-Sprint - team retrospective</td>
<td>E √ E E E</td>
<td>A-10 A-11 A-12</td>
</tr>
</tbody>
</table>

Table 8-10: Candidate ASD goals pursued by practices in ‘Post-Sprint’ method fragment (adapted from table 3-11)

As one would expect, the ASD goal of regularly reflecting on work and making improvements (A-12) is recognized by many team members in their consideration of goals that promoted post-sprint activities. Strong confirmation on the effectiveness of the retrospective practice (CS-11) to pursue and achieve this goal is expressed and certain team member’s contrast it very favourably with the more established ‘waterfall-based’ project post-mortem practice that has been a normal end of project process within the organization for many years.
“Much finer-grained than lessons learned...lessons learned that I have done over the past 10 years have just been once a year and was just ticking a box and did not change things”

Developer (SE)

What may not have been as obvious is the wide acceptance and understanding of the motivational (A-5) power of the sprint review (CS-10) practice. A very consistent awareness of the importance of this event was recounted by many of the developers, regardless of their core skill set. Statements that express a determination to succeed combined with a fear of failure portray this practice as a very motivational experience. They also reveal a strong collective responsibility within the sub-team that their particular work is perceived to meet expectations. The positive feedback experienced from success is also noted as delivering satisfaction to the team.

“There are times when it has been down to the wire for this... Something gone wrong and it becomes all hands on deck to fix it. It’s the ‘Limerick’ sprint review - not the code that Johnny wrote or Mary tested.”

Developer (QA)

“4 user stories failed...in 5 years time the team will still be talking about ‘black Thursday’ the day before the demo...The team were left looking at one another and saying that this can NEVER happen again”

Developer (BA)

“Having something at the end of 3 weeks that actually works...that you are showing to somebody and you are getting feedback on it ... praise I suppose on your completed work...is probably a motivating factor.”

Developer (SE)

8.4.5 Summary of ASD Findings

Table 8-11 summarizes the findings in relation to team members’ understanding of how their work practices may be motivated by ASD principles. This table offers a view of the landscape of comments that emerged in relation to the influence of different principles on the four identified method fragments of Scrum. However, the value of this research is in the idiographic details - not in a quantitative analysis of responses. These rich descriptions have been identified in the previous sections. The table provides visibility to outliers such as the large response to motivated individuals (A-5) as a goal of the post-sprint fragment. Although not addressed in the previous
section, the table reveals working software (A-7) as a strong influence also on post-sprint activities.

The table is presented in the same format as outlined in section 8.3.5. Asterisk and exclamation marks are used to indicate issues in relation to goal achievement, partial motivation and contradictions. Details of any contradictions are discussed in the earlier sections.
<table>
<thead>
<tr>
<th>GENERAL CODE Description</th>
<th>DScrum Goal</th>
<th>Practices</th>
<th>Gen</th>
<th>Pre-sprint Findings</th>
<th>Sprint Findings</th>
<th>Post-sprint Findings</th>
<th>ASD FRAMEWORK EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASD1 - Satisfy customer</td>
<td>g7</td>
<td>CS1; CS2; CS4; CS7; CS10; CS12</td>
<td>3*!</td>
<td>6</td>
<td>1</td>
<td>1*</td>
<td>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</td>
</tr>
<tr>
<td>ASD2 - Changing requirements</td>
<td>g8</td>
<td>CS2; CS4; CS6; CS7</td>
<td>0</td>
<td>5</td>
<td>2*!</td>
<td>0</td>
<td>Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.</td>
</tr>
<tr>
<td>ASD3 - Frequent software delivery</td>
<td>g9</td>
<td>CS7; CS10; CS12</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
</tr>
<tr>
<td>ASD4 - Business &amp; developer collaborate</td>
<td>g10</td>
<td>CS2; CS7</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>Business people and developers must work together daily throughout the project.</td>
</tr>
<tr>
<td>ASD5 - Motivated individuals</td>
<td>g11</td>
<td>CS4; CS7; CS8; CS9; CS10; CS12</td>
<td>0</td>
<td>3*!</td>
<td>4*!</td>
<td>8</td>
<td>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</td>
</tr>
<tr>
<td>ASD6 - Face-to-face commun.</td>
<td>g12</td>
<td>CS1; CS4; CS7; CS11</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</td>
</tr>
<tr>
<td>ASD7 - Working software is measurement of progress</td>
<td>g13</td>
<td>CS4; CS7; CS10; CS12</td>
<td>3*!</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>Working software is the primary measure of progress.</td>
</tr>
<tr>
<td>ASD8 - Sustainable Development</td>
<td>g14</td>
<td>CS4; CS6; CS7; CS8; CS9; CS11</td>
<td>0</td>
<td>8*</td>
<td>5*</td>
<td>1!!</td>
<td>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</td>
</tr>
<tr>
<td>ASD9 - Technical excellence and good design</td>
<td>g15</td>
<td>CS7; CS8; CS10</td>
<td>0</td>
<td>0</td>
<td>9*!!</td>
<td>3*</td>
<td>Continuous attention to technical excellence and good design enhances agility.</td>
</tr>
<tr>
<td>ASD10 - Simplicity</td>
<td>g16</td>
<td>CS2; CS4; CS6; CS7; CS8</td>
<td>1</td>
<td>3!</td>
<td>8*!!</td>
<td>0</td>
<td>Simplicity – the art of maximizing the amount of work not done – is essential.</td>
</tr>
<tr>
<td>ASD11 - Self-organizing teams</td>
<td>g17</td>
<td>CS4; CS10; CS11</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2*</td>
<td>The best architectures, requirements, and designs emerge from self-organizing teams.</td>
</tr>
<tr>
<td>ASD12 - team reflection and improvement</td>
<td>g18</td>
<td>CS10; CS11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6!</td>
<td>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.</td>
</tr>
</tbody>
</table>

Table 8.11: Overview of field results related to the presence of ASD goals

---

20 * indicates issues with goal achievement;  ! Indicates partial goal motivation;  !! Indicates goal contradiction
8.5 Findings associated with GSD Challenge Alleviation Goals

The third hypothesis proposed to structure exploration of this situation declared:

“A distributed team that uses Scrum will overcome the 3 identified GSD challenges”.

The structure used to explore this topic is a framework of GSD challenge alleviation goals constructed from earlier work in this field. Details of the framework are found in chapter four and the result of this chapter is a set of associations linking core Scrum practices to the pursuit of these goals (table 4-10). The three GSD challenge alleviation goals (GSD1 - GSD3) represent goals defined to overcome challenges in communication, coordination and control respectively. Each of these goal categories are further decomposed into challenges associated with different forms of distance (temporal, geographical and socio-cultural) in order to uncover practice-goal associations at a more granular level. These sub-categories of challenge alleviation goals are identified by a suffix indicating the distance challenge alleviated e.g. GSD1-T or GSD1-G.

Findings are presented in the same format as seen in the preceding two sections (8.3 and 8.4). The four sub-sections below present empirical research into the pursuit of various GSD challenge alleviation goals by practices performed within each Scrum method fragment. These findings are contrasted with the proposed findings presented in the candidate LDScrum model. Supporting tables use the same conventions to denote candidate and empirical associations.
### 8.5.1 General considerations

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>GSD Challenge</th>
<th>GSD1 Communication</th>
<th>GSD2 Coordination</th>
<th>GSD3 Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GSD1-T Temporal</td>
<td>GSD1-G Geographical</td>
<td>GSD1-C Socio-Cultural</td>
<td>GSD2-T Temporal</td>
</tr>
<tr>
<td>CS-1</td>
<td>Scrum Team Configuration</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>CS-2</td>
<td>Initial planning (and ongoing) - create or update product backlog</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-12</td>
<td>Definition of Done</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8-12: Candidate GSD Challenge Alleviation goals pursued by practices in ‘General Considerations’ method fragment (adapted from table 4-10)

The candidate model suggests that general Scrum considerations such as the team configuration and merging of roles tend to support achievement of all three challenge alleviation goals (GSD1 - GSD3). However, the empirical findings did not confirm such a comprehensive coverage of the various goals. The alleviation of coordination challenges (GSD2) was seen to be the main GSD goal pursued by the practices that comprised this method fragment. Mixed results emerged in relation to the efficacy of these Scrum practices to achieve this goal. In some instances, the Scrum of Scrums team configuration (CS-1) is endorsed as a practice that overcomes coordination challenges. Two respondents commented on the encapsulation of work resulting from this configuration leading to the Limerick team avoiding coordination complexity caused by team members in different time zones (GSD2-T).

“We are all sitting here in Limerick - we work wholly on our sprint whereas China are all in China working on their sprint”

Developer (QA)

However, one of those two respondents explained that although the Scrum of Scrums team configuration alleviates coordination complexity, it may also contradict the goal of alleviating geographical coordination issues that can cause a lack of awareness or reduced team spirit (GSD2-G).
“Before ... you would have folks from different locations on the same sub-team. ... with Scrum that cross-team integration is less...we are separate teams - if I did not have the QA meeting once a week I would not have interaction with the other teams like I would have had in the past”

Developer (QA)

From another perspective, a few respondents were quite non-committal on whether or not the general Scrum configuration and framework pursued GSD challenge alleviation goals. They seemed to indicate that any minor alleviation was just a side-effect of the structure and general practices.

“Initially, when we first started Scrum the fact that we were all working together, the communication seemed to go up a scale. It did improve morale. And in XX\(^2\) where they had very much worked in silos, working together in one group seemed to give them more interest and quite a lot of excitement”

Developer (SE)

8.5.2 Pre-Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>GSD1 Communication</th>
<th>GSD 2 Coordination</th>
<th>GSD3 Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-3</td>
<td>√</td>
<td>√</td>
<td>E</td>
</tr>
<tr>
<td>CS-4</td>
<td>√</td>
<td>√</td>
<td>E</td>
</tr>
<tr>
<td>CS-5</td>
<td>√</td>
<td>√</td>
<td>E</td>
</tr>
</tbody>
</table>

Table 8-13: Candidate GSD Challenge Alleviation goals pursued by practices in ‘Pre-Sprint’ method fragment (adapted from table 410)

Contrary to the proposed associations outlined in the above table, alleviation of GSD communication challenges was not noted as a goal for the performance of pre-sprint practices. Both the product owner and business analyst performed pre-sprint practices in order to overcome coordination challenges. The product owner felt that coordination complexity (GSD2-G) was successfully overcome by sprint planning although he did highlight that coordination among the distributed product owners was focussed primarily on the introduction of a new piece of functionality, not necessarily the start of each sprint. However, one can infer that such planning

\(^2\) XX: Location of Scrum team omitted to preserve anonymity
between the different product owners is performed to update the product backlog (CS-3) and present the prioritized items effectively to the team at the sprint planning meeting.

“We have 4 POs - one for each team obviously... that gives the broad approach within the financials team ... We agree what each team is going to do and that is where I talk with them. That is done at the start of a new piece of functionality (not a sprint).”

Product Owner

The business analyst recounted that during sprint planning, if he finds difficulty in grasping a particular concept, then being able to leverage the user story approach to discuss it with remote business analysts helps create a shared understanding of the concept (GSD2-T).

“If there is something that I just wanted more reassurance from other BA on other teams, I will send off an email...A 20 minutes talk-through can clarify up a lot of things - sharpen up the focus”

Developer (BA)

As with findings related to ‘general considerations’ (section 8.5.1), alleviation of the coordination challenge related to reduced team spirit (GSD2-G) is mentioned by some respondents. The view is taken that sprint backlog planning (CS-4) using story points might have contradicted this goal and that it did not alleviate the challenge.

“There is no training for what is a story point. We have got into the habit of breaking down into 3 point story. However we don’t know if this is equivalent to a 10 point story in another team.”

Developer (QA)

A difficulty around the application of story points is also brought up by another developer in their consideration of control issues related to the implementation of processes differently by distributed parties. The developer feels that the use of sprint planning practices such as the use of story points for estimation purposes (CS-4) is partially motivated by a desire to overcome different process usage across the teams (GSD3-T). However, in his opinion, this goal is not achieved.

“But if you look at the China, Antwerp, UK teams, they have different stories and I don’t know what tangible figure they associate it with.”

Developer (SE)
Alleviation of the challenges around different processes being used to manage product artefacts within a distributed team (GSD3-T) is addressed in a more positive light by the product owner. He felt that this challenge was pursued and achieved by sprint planning and he explained that regular sprint planning was always driven from a central backlog so this ensured that scope creep was avoided as any features not on the backlog were immediately discovered.

“We have a roadmap defined for our full year. So we know whether something might be off the roadmap”

Product Owner

8.5.3 Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>GSD Challenge</th>
<th>GSD 1 Communication</th>
<th>GSD 2 Coordination</th>
<th>GSD 3 Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td>E √</td>
<td>E</td>
<td>E √</td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td>√</td>
<td>√</td>
<td>E √</td>
</tr>
</tbody>
</table>

Table 8-14: Candidate GSD Challenge Alleviation goals pursued by practices in ‘Sprint’ method fragment (adapted from table 4.10)

The alleviation of communication challenges emerged as a goal being pursued by sprint practices. Alleviation of this challenge across all three distances (temporal, geographical and socio-cultural) was identified by respondents. The business analyst pointed out that the decomposition of work into clear simple user stories within sprints assisted in tackling time zone issues as emails can be targeted to a specific topic and this helped to reduce delays that might have arisen due to misunderstandings (GSD1-T) related to asynchronous communication (such as an email).

“My boss is in China. If I run into a problem at 10 a.m. I can send him an email... You are asking very specific questions - there is no wishy-washy. You are working at a very granular level.”

Developer (BA)
Overcoming the challenge of increased effort to initiate contact (GSD1-G) among remote parties was alleviated by the establishment of a weekly status meeting by product owners (CS-7). This meeting ensured that remote parties could rely on the other attendees to be present in order to address various issues across the distributed project.

“We meet every Thursday with the product owners and if BA has brought anything back we can review and agree at that point...meetings are held weekly for escalations, etc.”

Product Owner

In relation to socio-cultural distance, the business analyst noted that a key criterion he applied when documenting regular sprint activities was that the material could be read by a non-native English speaker. Along the same lines, another developer explained that they have a regular weekly meeting which is attended by a technical representative from each of the distributed teams (CS-7). Overcoming different frames of reference (GSD1-C) is achieved because the material being discussed falls within the remit of one sprint and is granular enough for a common understanding to emerge within the meeting.

“That is true (that working in 3 week sprints helps distributed meetings). I am on the tech rep meeting and Scrum is helpful for people to discuss what is happening in those 3 weeks.”

Developer (SE)

Various coordination challenges were pursued and achieved by the performance of sprint practices. The alleviation of reduced team spirit and lack of awareness (GSD2-G) was managed through the use of central development tools such as Atlassian JIRA and Greenhopper. Developers could access the activities of remote teams on these tools.

“If you feel that you are having problems with something, I would look to see the way that they are doing it. It might work better for us... looking in JIRA and Greenhopper and how they do things in there.”

“Use Greenhopper which links with JIRA (bug tracker), code manager (subversion) and we use that to track our sprint progress mainly to give people outside the office visibility to where we are.”

Developers (SE)

---

22 BA: Business Analyst’s name removed to preserve anonymity
In response to a query on how the Limerick team might become aware of work being performed in another team that could impact their activities, the product owner claimed that he was responsible for this coordination. However, reliance on one individual to manage coordination complexity is one of the challenges associated with GSD. Placing this responsibility solely on the product owner does not overcome this issue. However, various respondents referred to other meetings performed within the sprint (CS-7) as practices that alleviated coordination complexity (GSD2-G). The weekly ‘Scrum of Scrum-masters’ is noted as an effective practice used to alleviate coordination complexity.

“The Scrum-master reviews...gives an insight into what the other teams are actually working on and you know if they are working on a similar piece of code that you are and that is going to affect you.”

Developer (SE & Scrum-Master)

Another central coordination meeting used was a weekly architect’s meeting. Although the primary focus of this meeting was to address design issues, a medium goal of this practice is likely to have been to overcome the coordination challenge of a lack of mechanisms for creating shared understanding (GSD2-G). In dealing with design concepts, such mechanisms are very useful in discussing abstractions which form a major element of the design process. However, an observation made in relation to this meeting is that it did not always achieve this goal.

“Inter-Scrum technical communication... we do have that architects meeting once a week which has varying degrees of success. It depends on how much somebody goes into what they are doing”

Developer (SE)

Various different meetings were summarized by one software engineer as practices that are intended to overcome coordination complexity (GSD2-G). His response appears to indicate knowledge of coordination activities but little appreciation for such work.

“I don’t know what the overall team does to coordinate work. Scrum masters meet ... tech representative meeting weekly. There is a business rep meeting. There is also a PO meeting”

Developer (SE)
Control challenges are pursued by Sprint practices. The Scrum-master observed that the management of project artefacts between distributed teams (GSD3-T) was pursued and achieved as general sprint activities incorporated effective documentation and recording of activities.

“Documentation is updated more often - rather than leaving it to the very end of the year when it has to be done and ... trying to figure out “what did I do back then?”...We do it as we go along”

Developer (SE & Scrum-master)

The burn-down chart (CS-9) was used in order to alleviate the challenge of controlling a project across distributed teams (GSD3-T). In this situation, it was used mainly by management so although the team may not have had a lot of regard for this device, it was a technology that was perceived by management to be useful for their needs.

“When we go into the Scrum-master meeting, PM\textsuperscript{23} would be looking at the chart a lot and seeing how our work is going. I would not be looking at the burn-down chart as much.”

Developer (SE & Scrum-master)

Collaboration between product owners on how to address an emergent requirement alleviated the challenge of controlling distributed development. A lack of concurrent engineering principles (GSD3-G) could lead to additional functionality costing more than it is worth. However, a clear reporting mechanism alleviated this challenge. Such effective governance is evident from the product owner’s response to a query on whether he would block development of additional functionality within a sprint if it was suggested by the business analyst.

“...not only just in the Sprint - but (I would) also (check) in our PO meeting on Thursday to see if things are really worth doing (from a cost perspective)”

Product Owner

\textsuperscript{23} PM: Overall project manager - name removed to preserve anonymity
8.5.4  Post-Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Scrum Challenge</th>
<th>GSD1 Communication</th>
<th>GSD 2 Coordination</th>
<th>GSD3 Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-10</td>
<td>Post-Sprint - review meeting</td>
<td>GSD1-T Temporal</td>
<td>GSD1-G Geographical</td>
<td>GSD1-C Socio-Cultural</td>
</tr>
<tr>
<td>CS-11</td>
<td>Post-Sprint - team retrospective</td>
<td>√</td>
<td>E</td>
<td>E√</td>
</tr>
</tbody>
</table>

Table 8-15: Candidate GSD Challenge Alleviation goals pursued by practices in ‘Post-Sprint’ method fragment (adapted from table 4 10)

Alleviation of Coordination challenges (GSD-2) was the main goal to emerge from enquiries into post-sprint practices. Rules and processes were established to manage implementation of the post-sprint review (CS-10) in order to overcome certain challenges associated with having parties distributed across different time zones. Although there were four teams, the sprints were staggered so that only two sprints were ever completed on the same day. This enabled effective time management so that on the review day, the two completing teams can conduct their reviews within meeting schedules that account for reduced hours of collaboration due to time zone restrictions (GSD2-T).

“They have their sprint review on a Friday evening 4 - 5 which is 9 to 10 a.m. for us and 10 - 11 for Antwerp. We have put in rules for review hours to manage this.”

Developer (SE)

Even if individuals were unable to be present, reviews (CS-10) were recorded in order to overcome the difficulty of synchronized team meetings (GSD2-T). Although not specifically highlighted by any respondent, another potential challenge that may be overcome by this practice is communication misunderstandings due to language difficulties (GSD1-C). If a team member has difficulty comprehending the demonstration at the time of delivery, they will have the option to replay the recording at a pace more suited to their needs.
“The recordings are good so long as you keep them time-bound and structured. The team introduces and
gives their demo and takes questions.. if somebody is stuck for time, they know what parts of the recording
to listen to”

Developer (SE)

A number of respondents felt that the sprint review provided a mechanism for creating shared
understandings (GSD2-G). The product owner’s observations on this describe his interpretation
of the effect of this practice.

“Everybody goes to the Sprint demo so we get to see what the others are doing and also what they are
working on for the next 3 weeks...Everybody should know what everybody else is working on.”

Product Owner

Although this interpretation of the review practice is endorsed by another developer, there were
other responses that were not quite as assured on the efficacy of the practice with regard to this
goal. One respondent explained how the need to share abstractions (GSD2-G) was not being met
by the review and so a number of meetings emerged to address this issue.

“In the review...the technical review was omitted because of the audience. That was why the idea came to
have teams bring their work to the architects meeting but (all the) developers are not there....”

Developer (SE)

Although the sprint retrospective (CS-11) was considered in a very positive light when Limerick
team members discussed learning and growth within their own sub-team, there was no evidence
of insights generated from this practice being leveraged across the wider team. One developer
reflected that increasing team spirit and awareness (GSD2-G) could be a motivator for pursuing
the expansion of the retrospective practice to share insights between teams. However, he
admitted that this practice was not being performed.

“Perhaps we should be sharing things that we find very good. We have not done that to date. Whether or
not that is being shared at the Scrum-master meetings, I don’t know.”

Developer (SE)
An interesting observation noted that the sprint review could actually be divisive and instead of overcoming the challenge of reduced team spirit (GSD2-G), it may compound the distinctions between different remote sub-teams.

“Demos help focus and they may help get all 40 person team together, but I think that Scrum has helped each different sub-team have identity - I’m not sure about pulling all sub-teams into overall team.”

Developer (SE)

8.5.5 Summary of GSD Findings

Table 8-16 presents an overview of responses supplied in relation to practices that are motivated by a need to overcome GSD challenges. The notation used in the other findings overview tables (table 8-6Table 8-11) is applied to this table also. An asterisk denotes issues with goal achievement and an exclamation mark highlights findings where one or more practices may have only partially motivated a goal. An interesting point that emerged across a number of the method fragments was that the application of certain practices within the Scrum of Scrums configuration may contradict the goal of alleviating reduced team spirit. The promotion of a sense of responsibility and autonomy within a Scrum sub-team may inadvertently lead to increased independence between these teams and consequently raise the lack of awareness and lack of team spirit within the overall distributed team. Such contradictions are denoted by two exclamation marks (!!).

This table reveals that respondents more readily identified with practices that alleviated coordination challenges (GSD-2). Of 55 responses, just 4 related to communication (GSD-1) and 7 were associated with control (GSD-3). Of the four method fragments explored, the majority of responses (27) relate to sprint practices and 15 to post-sprint, leaving just 13 observations from the other two fragments.

This analysis focussed on the alleviation of challenges associated with three GSD processes. Additional detail was identified by the categorization of findings according to the type of distance impacted by each GSD process. This additional granularity was used to support the accuracy of the analysis and also to deepen the description of how practices motivated the
alleviation of GSD challenges. Chapter 5 explained how alleviation of the three higher-level challenges is used to extend the LDScrum model. Therefore, when building the refined LDScrum model, all findings are synthesized to the higher-level GSD processes. For example, if a response indicated that a sprint practice was performed to overcome language difficulties, this would be described as GSD1-C (alleviation of a communication challenge resulting from socio-cultural distance) but would be reported in LDScrum as a communication challenge alleviation goal. The representation of NPD, ASD and GSD goals as elements of the LDScrum model is explained at the outset of the next section.
<table>
<thead>
<tr>
<th>CODE</th>
<th>DScrum Code</th>
<th>Practices</th>
<th>Gen</th>
<th>Pre-sprint</th>
<th>Sprint</th>
<th>Post-sprint</th>
<th>Specific challenge</th>
<th>GSD Framework Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD1-T</td>
<td>G19</td>
<td>CS7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Delayed Communication</td>
<td>It may take longer to contact somebody when they are needed. There may be delayed response due to misunderstandings and it may take longer to resolve entire issue due to delayed feedback.</td>
</tr>
<tr>
<td>GSD1-G</td>
<td>G19</td>
<td>CS7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Lack of informal communication</td>
<td>Reduced chance of informal communication - reduced opportunity to build working relationships. Reduced flow of info. on changes and loss of essential aspect of design activities. High dependency on effective tools and environments. May be variances in available technology. Increased costs due to necessity for travel. Difficult to determine remote skillsets and who to contact.</td>
</tr>
<tr>
<td>GSD1-C</td>
<td>G19</td>
<td>CS7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Language Differences and misunderstandings</td>
<td>Language misunderstandings. Also need to overcome different frames of reference in communication.</td>
</tr>
<tr>
<td>GSD1</td>
<td>G19</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td>Total = 4</td>
<td></td>
</tr>
<tr>
<td>GSD2-T</td>
<td>G20</td>
<td>CS10</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Reduced Hours of collaboration</td>
<td>Misunderstandings &amp; rework can cost. Reduced hours of collaboration. Difficult to have synchronous team meetings. Downtime of needed technologies Coordination can be costly due to amount of non-routine work - needs close management.</td>
</tr>
<tr>
<td>GSD2-G</td>
<td>G20</td>
<td>CS1:CS10;CS11:CS7;CS9</td>
<td>2*!!</td>
<td>4*</td>
<td>2*</td>
<td></td>
<td>Lack of awareness / team spirit</td>
<td>Reduction of trust Lack of team spirit - can lead to conflicts - also lack of knowledge of availability of remote colleagues. Teams may not realize critical tasks without good information sharing mechanisms.</td>
</tr>
<tr>
<td>GSD2-C</td>
<td>G20</td>
<td>CS12; CS7</td>
<td>1*!</td>
<td>2</td>
<td></td>
<td></td>
<td>Lack of domain knowledge</td>
<td>May have doubts of their colleagues abilities May have conflicting views of domain knowledge May require language and cultural training.</td>
</tr>
<tr>
<td>GSD2</td>
<td>G20</td>
<td>CS10</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Doubtful of other’s abilities</td>
<td></td>
</tr>
<tr>
<td>GSD2</td>
<td>G20</td>
<td>CS1; CS10;CS4:CS7</td>
<td>2*!!</td>
<td>2*!!</td>
<td>2</td>
<td>1*!!</td>
<td>Lack of awareness / team spirit</td>
<td></td>
</tr>
<tr>
<td>GSD2</td>
<td>G20</td>
<td></td>
<td>7*!!</td>
<td>4*!!</td>
<td>20*!!</td>
<td>13*!!</td>
<td>Total = 44</td>
<td></td>
</tr>
<tr>
<td>GSD3-T</td>
<td>G21</td>
<td>CS3:CS4:CS7</td>
<td>2*!</td>
<td>2</td>
<td></td>
<td></td>
<td>Management of project artefacts</td>
<td>Differing processes may be hard to manage.</td>
</tr>
<tr>
<td>GSD3-G</td>
<td>G21</td>
<td>CS7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Lack of concurrent engineering principles</td>
<td>Issue with completion of processes between sites. Requirements mods, poor tools, lack of informal contact impact concurrent engineering.</td>
</tr>
<tr>
<td>GSD3-C</td>
<td>G21</td>
<td>CS10</td>
<td>2!</td>
<td></td>
<td></td>
<td></td>
<td>Perceived threat from low-cost alternatives</td>
<td>Perceived threat of loss of job to lower-cost colleague Need to be aware of local norms and practices Different perceptions of authority.</td>
</tr>
<tr>
<td>GSD3</td>
<td>G21</td>
<td></td>
<td>0</td>
<td>2*!</td>
<td>3</td>
<td>2!</td>
<td>Total = 7</td>
<td></td>
</tr>
</tbody>
</table>

Table 8-16: Simplistic view showing field results for presence of GSD challenge alleviation goals

* indicates issues with goal achievement; ! indicates partial goal motivation; !! indicates goal contradiction
8.6 Findings associated with Lean Software Development Values

A set of candidate LSD value to goal associations was presented in chapter 5. These associations were defined based upon an analysis of each NPD, ASD and GSD goal description in order to determine if the LSD principles associated with a particular value would underpin the goal. A hypothesis is proposed to frame the analysis of this inquiry:

“The presence of all 12 identified Lean Software Development (LSD) values will be evident in projects that are performed by distributed teams that use Scrum”

This analysis resulted in the presentation of a Scrum value rationale from both an external (figure 5-11) and internal (figure 5-12) orientation. These value rationales are presented in graphical format in the presentation of three method fragments of the overall LDScrum model in chapter 5 (figures 5-13, 5-14 and 5-15). In order to assist in the comparison of the candidate value rationale to the empirical findings, table 8-17 below outlines the associations of the combined internal and external value rationales in tabular format. The rest of this chapter describes how the empirical findings are examined in order to refine the proposed value rationales to reflect interpretations of the social reality at play in the Fin-Link team’s performance of their work. This section concludes with an outline of the refined value rationale (table 8-32).

Please note that the previous sections (8.3, 8.4 and 8.5) used the prefix ‘NPD-n’, ‘A-n’ and ‘GSD-n’ to denote NPD, ASD and GSD goals respectively. This section uses the mapping of these goals to their LDScrum goal code (g1 - g21) as portrayed in table 8-17.
### LDScrum Values

<table>
<thead>
<tr>
<th>LDScrum Goal</th>
<th>Code</th>
<th>Description</th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
<th>V7</th>
<th>V8</th>
<th>V9</th>
<th>V10</th>
<th>V11</th>
<th>V12</th>
</tr>
</thead>
<tbody>
<tr>
<td>g1</td>
<td>NPD-1</td>
<td>Built-in instability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g2</td>
<td>NPD-2</td>
<td>Self-organizing project teams</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g3</td>
<td>NPD-3</td>
<td>Effectively Overlap development phases</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g4</td>
<td>NPD-4</td>
<td>Multi-level learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g5</td>
<td>NPD-5</td>
<td>Leverage Subtle control over work tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g6</td>
<td>NPD-6</td>
<td>Disseminate learning through organization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g7</td>
<td>A-1</td>
<td>Satisfy the customer: early val. software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g8</td>
<td>A-3</td>
<td>Welcome changing requirements,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g9</td>
<td>A-2</td>
<td>Deliver working software frequently</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g10</td>
<td>A-4</td>
<td>Business and developers work closely.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g11</td>
<td>A-5</td>
<td>Build projects around motivated individuals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g12</td>
<td>A-6</td>
<td>Face-to-face communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g13</td>
<td>A-7</td>
<td>Working software as measure of progress.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g14</td>
<td>A-8</td>
<td>Sustainable development.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g15</td>
<td>A-9</td>
<td>Continuous attention to technical excellence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g16</td>
<td>A-10</td>
<td>Simplicity.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g17</td>
<td>A-11</td>
<td>Self-organizing teams.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g18</td>
<td>A-12</td>
<td>Regular Team reflection.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g19</td>
<td>GSD-1</td>
<td>Overcome GSD communication challenges</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g20</td>
<td>GSD-2</td>
<td>Overcome GSD coordination challenges</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g21</td>
<td>GSD-3</td>
<td>Overcome GSD control challenges</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8-17: Summary of candidate LDScrum value rationale

### 8.6.1 ‘V1’ Customer Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - G7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer value</td>
<td>Ensure customer defines value. Seek out user needs, not just requirements. Early delivery of value. Enable future needs (maintainability). Enable efficient deployment of product features. Provide value for money.</td>
<td>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</td>
</tr>
</tbody>
</table>

Table 8-18: Candidate Value Rationale for V1 - ‘Customer Value’
The table above presents the goals that have been identified as candidates to be pursued as a result of the underlying LSD value of Customer Value (V1). Analysis of empirical data indicated that customer satisfaction (g7) was indeed pursued as a result of this value. However, additional goals that appeared to be influenced by V1 were:

- Delivery of working software frequently and preferably early (g9)
- Regular collaboration between developers and business people (g10)
- Motivated and trusted developers given suitable environments (g11)

Two findings identified within the post-sprint fragment displayed aspects of this value in the views expressed by certain software engineers about the sprint review. They noted that this practice pursued the goal of early delivery of working software (g9) which in essence should constitute early delivery of value. This is an identified principle of V1 and as such, forms part of the description of this value in table 8-18 above. Frequent software delivery is pursued by completing software each sprint and then organizing merge sprints (due to environment complexity) to release to the next level of proximity to the customer. This efficient deployment of product features is underpinned by Customer Value (V1).

“You have to have s/ware that you can demo. We check our code into our Limerick environment, and we merge it into a FIN-NN trunk ... You would work in the Limerick environment for a few sprints and then you would merge it in.”

Developer (SE)

As one would hope, interpretations of some reflections supplied by the product owner reveal an underlying value of V1. He referred to the boundary of a three week sprint as being helpful in achieving a goal of using motivated people (g11). Having motivated people in this instance is a goal that is underpinned by the importance of efficient deployment of product features that the user needs - Customer Value (V1). One could argue that this is a general observation about all practices, but when pressed, the product owner identified the Pre-Sprint work (which involves the selection of product features) to be key to the achievement of this goal.
“I see huge value in using these phases ... the 3 weeks focus the minds of the people and highlights importance of the work ... Pre-sprint work is most valuable - you are setting your priorities.”
Product Owner

The Product owner explained that as a participant in sprint practices, he represented the business. In essence, he pursued a goal that business people and developers work together daily throughout the project (g10). In his opinion, his constant availability to answer questions and clarify requirements issues helped to ensure that user needs and value for money were a foremost consideration in development decision-making. The provision of value for money is a key aspect of V1 and as a result it appears that g10 is underpinned by V1.

“I am there to answer questions and make decisions. If the team come up with the approach and I don’t like it I would say no. My responsibility is to deliver to the wider team. How far do we go into something?”
Product Owner

However, it should be noted that the majority of findings do ratify the candidate value rationale proposing that satisfying the customer (g7) is underpinned by V1. A general observation by the product owner in relation to escalations described how the insertion of escalations into sprints delivered valuable important software to the customer early. His emphasis on speed and early delivery indicates that V1 is the underlying value that influences the pursuit of this goal.

“We have delivered early in numerous occasions by dealing with escalations.... in the case of escalations we can do work that is patched to the customer very quickly.”
Product Owner

Many findings that pursued the goal of early delivery of valuable software emerged from reflections of the pre-sprint method fragment. An excerpt from the business analyst illustrates how customer focus influences the pursuit of the selection of valuable software.

“We tried to maintain our customer focus a lot by using our PO....the PO comments would be key to whether it was acceptable or not.”
Developer (BA)
8.6.2 ‘V2’ Reduce Waste

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce Waste</td>
<td>Promote systematic defect prevention. Eliminate rework through emphasis on customer needs. Holistic focus on quality that motivates the refactoring of legacy code to improve general product. Identify silos of waste present in value stream and improve processes. Promote reusability where appropriate. Take cognisance of the fact that waste can have different forms in different projects/work situations.</td>
<td>Simplicity - the art of maximizing the amount of work not done - is essential</td>
</tr>
</tbody>
</table>

Table 8-19: Candidate Value Rationale for V2 - ‘Reduce Waste’

The candidate value rationale proposes that just the pursuit of simplicity (g16) is the only goal underpinned by the value of waste reduction (V2). The product owner indicated that he viewed waste reduction as a key influence in his pursuit of simplicity. He described his role in meetings as that of a central expert that integrates overall development. This would suggest that the value at play here is effective process (V10). However, his explicit comment with regard to the avoidance of ‘gold plating’ (PMI, 2008) highlights that waste reduction (V2) appears to be the dominant value here.

“The team can overcomplicate things and do more than is needed. That is why you need to have a business view so that you can balance the customer’s requirements against the cost of doing something”

Product Owner

Many of the reflections prompt the likelihood of certain other goals being underpinned by V2:

- Subtle Control (g5)
- Welcome changing requirements (g8)
- Delivery of working software frequently and preferably early (g9)
- Working software is the primary measure of progress (g13)
- Continuous attention to technical excellence (g15)
The product owner noted that he needed to take care in his approach to the control of team activities. He exhibited subtle control (g5) by allowing the team determine the work that they would perform within limits but he would intervene to control them if necessary. In this excerpt, such intervention is prompted by a value of waste reduction (V2).

“I felt we were doing too much functionality for one or two customers. You have to balance business needs with the cost of the work.”

Product Owner

An interesting counterpoint by one software engineer indicated that a value of waste reduction motivates them to consider not pursuing simplicity (g16). Although he accepted that in his regular work he strove to achieve simplicity, he did not like being compelled to keep things simple. He indicated that pursuit of this goal may contradict the pursuit of technical excellence (g15). He felt that pursuing simplicity can be counterproductive and that any potential future needs/fixes should be raised as issues. This attention to the notion of future proofing your work indicates aspects of the value of ‘Business Environment Awareness’ (V8). Alternatively, one could infer that he wished to deliver functionality in change tolerant form: ‘Product Excellence’ (V6). However, his example suggests that his main concern is that when defects are uncovered they should be resolved - which would indicate an influence of waste removal (V2).

“I have to ignore things that I see that I can improve. If a car mechanic went to fix your brakes and he spotted something wrong with the clutch should he ignore it. It’s a tunnel view”

Developer (SE)

This view is echoed by the business analyst who revealed that although the pursuit of simplicity is performed by the team (in sprint planning and other meetings), such pursuit can contradict a goal of good design (g15) and as such can dilute or ignore the value of waste reduction (V2). This may occur due to a failure to promote the refactoring of legacy code to improve general product, thus allowing technical debt to accumulate.

“This can go against resolving technical debt. As the Scrum-master training said, no new technical debt”

Developer (BA)
The NPD goal of self-organization (g2) is identified as one of the goals pursued by post-sprint practices (section 8.3.4). The particular finding referred to a situation where the team moved their collocated retrospective back to the original time of the last day of the Sprint. They had previously moved that internal sub-team practice to the first day of the next Sprint in order to best meet the availability of all team members. However, the following excerpt presented by the business analyst explains how his goal to self-organize was underpinned by a value to reduce waste caused by the dilution of momentum.

“If you start doing that (Sprint Retrospective) on a Monday, it will eat up an hour of energy that could be better spent on that post-match euphoria that you have on the Friday.”

Developer (BA)

The developer whose core skill is quality assurance specialist observed that pursuit and achievement of the frequent delivery of working software (g9) reduced the need to manage artefacts that would normally be necessary for longer development cycles. Reduction of management waste resonates with the findings of (Middleton et al., 2005).

“We used to have a test case matrix... you had different builds at different times and different levels of completion of the test suite. With Scrum, you do a sprint and introduce a piece of functionality and your tests are complete - there is no need for a matrix.”

Developer (QA)

Waste reduction was seen as the underlying value in observations made in relation to the pursuit of working software as the primary measure of progress (g13). These reflections came from developers with different core competencies (QA, BA and SE).

The restriction that the ‘demonstration’ (CS-10: sprint review) would only present functions that are deemed complete pursues the goal of highlighting working software as the main measure of productivity (g7). This goal is underpinned by the value of waste removal as you seek to omit silos of waste (such as defects).
“You have to have it working. If it is not working you don’t show it. Everything must be finished - all coding, all testing, all defects closed. Every task associated with a story must be complete”

Developer (QA)

Putting in an appropriate amount of work indicates an awareness of waste. The application of either too much or too little effort is undesirable. The promotion of working software as the primary measure of progress (g7) promotes effective approaches to the completion of tasks and is underpinned by the value of removing waste that lurks in managing multiple unfinished deliverables.

“The demo (ensures that working software is our main measure of progress) ... You don’t want to over prepare for the demo and we have got much better at putting in the right amount of prep work”

Developer (SE)

The following excerpt further describes how the demonstration serves to focus the minds of the team to meet their goal. It seems as if the goal of working software being the primary measure of progress (g7) supports the goal of motivated individuals (g5). This particular observation describes the waste associated with generating momentum to tackle a piece of work that was not completed within a sprint and had to be carried forward into the next sprint. In the developer’s opinion, if the work had been completed within the sprint, this waste would have been avoided.

“...even though on the previous Monday we .. expected to have it completed for Friday and we .. worked on it for four days, I have never seen that type of story complete on the following sprint without taking up another 4 days. We just seem to lose that momentum ..it takes a long time to get going again.”

Developer (BA)

Seeking to promote sustainable development (g14) also appeared to be concerned with waste reduction. Confining the required work to a distinct set of activities aids in the pursuit of sustainable development. One developer felt that this goal is achieved by the early identification of defects and incorrect solutions which indicates an underlying value of waste reduction.

“You can see obstacles a lot quicker with working in 3 week sprints. You can see if you are not going to hit a target early on in a Sprint...The smaller parts of work help you to find the problems earlier.”

Developer (SE)
Continuing on the theme of ‘smaller parts of work’, the BA comments on the promotion of sustainable development (g14) through the decomposition of work into very clear acceptance criteria. This leads to effective and efficient generation of dependent work (such as test creation), therefore leading to a reduction in the waste associated with inefficient work practices.

“Work on a granular approach. Start the sentence and get to the full stop without using ‘and’ - a clear concise thing...getting those acceptance criteria at that point makes creating their tests very quick.”

Developer (BA)

### 8.6.3 ‘V3’ Flow of Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g9</th>
<th>Goal - g14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow of Value</td>
<td>Stabilize development process to enable levelled flow of value - manage necessary variability. Smooth demand from users. Pull value from user demands through lifecycle. Apply minimalism (scope items, teams, documents). Allocate resources only when needed.</td>
<td>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
<td>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely</td>
</tr>
</tbody>
</table>

Table 8-20: Candidate Value Rationale for V3 - ‘Flow of Value’

Initial analysis proposed that ‘frequent delivery of working software’ (g9) and ‘sustainable development’ (g14) are the two goals underpinned by a concern for ‘Flow of Value’ (V3). There is just one reflection indicating that pursuit of the frequent delivery of working software was performed due to this value and the value-goal relationship is assessed as being quite weak for that particular finding. However other findings provide interpretations that support a firm relationship between V3 and G14. Furthermore, a relationship between the pursuit of a number of other goals and this value emerged - especially in discussions around sprint activities:

- Self-organization (g2)
- Welcome changing requirements (g8)
- Delivery of working software frequently and preferably early (g9)
• Working software is the primary measure of progress (g13)
• Simplicity (g16)

The explanation linking frequent delivery of working software (g9) to flow of value (V3) was described in the context of the definition of done (DOD) core practice (CS-12). This explanation indicates that at that point in time, the team’s DOD is at a simple level. There appeared to be some indication of a regard for the regular flow of value to a customer but the analyst deemed the relationship to be weak in this instance. However, if such a regard for the flow of customer value (V3) is present within the team, it may motivate them to intensify their pursuit of the frequent delivery goal (g9) by increasing the requirements of their DOD as their application of Scrum matures. It is worth noting that a concern with low levels of DOD is that they can lead to an increased level of technical debt, a topic that has emerged in other findings.

“It makes sense to produce in a couple of weeks...You build on from sprint to sprint. I am not sure what we produce at the end of each sprint is fully working... I am not sure if you could give it to a customer.”

Developer (SE)

A common theme that emerged in relation to the pursuit of G14 (sustainable development) was the application of minimalism to the definition of scope. This theme supported a need to stabilize the development process in order to sustain development by producing a stream of clear outputs. This objective appears to be based on an appreciation of the need to maintain a steady flow of value.

“The story points are high-level. We see the big stories and we break them into smaller stories. We may find we won’t get them all done so it’s back to ... determine what are high-priority and focus on them”

Developer (QA)

As one would expect, the concept of minimalism is evident in the pursuit of simplicity (g16). Findings across the method fragments produced interpretations of this goal being underpinned by flow of value (V3). Sprint planning (cs-4) was modified to reduce potential complexity. As a consequence of issues within Sprints, user stories were tightened up to deal with the standard situation initially, followed by separate stories for exceptional cases. This achieved the goal of
simplicity and although it emerged from reflections on defective sprints, the modification of the practice to pursue this goal is underpinned by the value to apply minimalism and manage necessary variability which is one of the principles associated with flow of value (V3).

“We thought it was not too bad but when we got into it, it was a way bigger than we thought... Every sprint after that, we broke out stories so that we did anything for ‘normal’ first then ‘staged’ second.”

Developer (QA)

Pursuit of simplicity (g16) is also found in sprint activities. One developer explained how the core practice of locking features within a sprint was pursued. The pursuit of simplicity in order to maintain clear scope within a sprint is interpreted as an indication of the presence of a value to maintain flow from sprint to sprint.

“We have got better at locking features. We do make a choice at our daily meetings to determine if something is part of our task or should it be on the backlog. We have got much stricter about this”

Developer (SE)

A related finding highlighted that while the practice of locking features (CS-6) within a sprint is desirable, it needs to be tempered by the realization that it may contradict the goal of welcoming changing requirements (g8). However, in the specific reflection provided by the product owner, his concern for this contradiction is somewhat diluted by the view expressed that unlocking the sprint is not desirable, but sometimes necessary. It appears to reveal an underlying value to manage necessary variability.

“Well if I get an escalation I do (change priority during the sprint). It’s the real world after all. It’s grand to say the team should be buffered but you have to deal with escalations.....I don’t like doing it”

Product Owner

‘Embracing change’ (V7) is the value that one would naturally associate with the goal of welcoming changes (g8). However, one developer’s reflections reveal flow of value (V3) as an influence on this goal. His acceptance of how the product backlog may evolve from customer
requests showed a willingness to embrace change and ensure continual flow of value to the customer.

“Our backlog comes from enhancement requests or functionality that we needed. Then as time goes on it expands. It is a living thing...”

Developer (SE)

Pursuing the measurement of progress through working software (g13) has also been underpinned by V3. In the following excerpt, stabilization of the development process to enable levelled flow by smoothing demand indicates V3 as the key value underpinning achievement of this goal.

“At one time we thought of putting coding and test phases because nobody ever know whether it was in test or not. Now we are saying that only 2 user stories can go through at any one time.”

Developer (SE)

The pursuit of behaving as a self-organizing team (g2) is described through the team’s application of self transcendence in relation to story point allocations. The description appears to infer that the goal is pursued in order to smooth demand and enable a levelled flow of value.

“If something has too high story points we consider it too big to do in 3 weeks. This is being learnt from experience.. we used to get pushed a bit to do more and then we settled ... we commit to so much and if we get that done we may add something from the backlog.”

Developer (QA)

### 8.6.4 ‘V4’ Person Focus

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g2</th>
<th>Goal - g5</th>
<th>Goal - g11</th>
<th>Goal - g17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person Focus</td>
<td>Empower individuals and teams. Promote continual worker development. Invest in multi-skilling and promote attitude to participate in multiple roles.</td>
<td>Self-organizing project teams. (Autonomy, Self-transcendence, Cross-fertilization)</td>
<td>Leverage Subtle control over work tasks</td>
<td>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</td>
<td>The best architectures, requirements, and designs emerge from self-organizing teams</td>
</tr>
</tbody>
</table>

Table 8-21: Candidate Value Rationale for V4 - ‘Person Focus’
Findings indicated that all four candidate goals were underpinned by person-focus (V4). Three additional NPD goals and one ASD goal were also influenced by this value:

- Built-in Instability (g1)
- Overlapping Phases - collective responsibility (g3)
- Multi-level Learning (g4)
- Sustainable Development (g14)

As outlined in the description provided in table 8-21 above, a core aspect of ‘Person Focus’ relates to team empowerment and this value is evident in reflections on the pursuit of many of the goals. Various practices have been identified in section 8.3 that reflect on the team’s pursuit of ‘Built-in Instability’ (g1). The pursuit of this goal involves setting challenging targets and releasing control to the team on how they achieve those targets. Releasing control empowers the team which clearly is underpinned by ‘Person Focus’. For example, in the following excerpt the developer describes a situation where the team is clearly empowered to address a requirement.

“Yes - it could happen (that the PO might question an estimate). But he would have to concede…..The software engineer would say what it is and generally - you can’t argue with them. They know the code.”

Developer (BA)

In relation to the distributed Scrum of Scrums team configuration, the framework provides autonomy to each of the sub-teams, thus contributing to the pursuit of self-organization (g2). However, the excerpt explains that the success or usefulness of this goal within a sub-team is contingent on the team’s ability to work without external support or interference. The interpretation is that the pursuit of such a goal within this context reveals an underlying value to empower individuals and teams which would indicate person-focus (V4).

“I accept that you are giving more autonomy to the teams but this depends upon the team’s ability. The core knowledge still resides in XX25.... We want to have sharing between teams also”

Developer (SE)

25 XX: Team name (location) omitted to preserve anonymity
The goal of overlapping development phases (g3) is pursued by developers executing tests. The value underpinning this goal is to promote an attitude to participate in multiple roles.

“We try to work on any of the tasks that are available even if it is not something that you normally do”
Developer (SE)

One particular developer described how his participation in acceptance criteria meetings helped him to pursue the goal of learning across functions (g4). The promotion of continual worker development is one of the principles that form the description of person-focus (V4).

“So everything is new to me and I am learning constantly from the team and that is why those acceptance criteria meetings are so useful to me.”
Developer (SE)

The general Scrum structure enables the team to determine how best to tackle their work. It supports the establishment of set boundaries that clarify to the team what must be produced. The pursuit of this goal of subtle control (g5) is strongly influenced by team empowerment indicating an underlying value of V4.

“In a Scrum team the PO is the customer... He is not concerned how we do something - just that we produce”
Developer (SE)

The pursuit of motivating individuals (g11) was identified as a candidate goal that may be underpinned by person focus. The post-sprint method fragment and more specifically, the core practice of sprint review (CS-10) is found in many instances to pursue and achieve this goal (see section 8.4.4 above). The collective ownership evident in various excerpts that describe the motivational (g11) aspects of sprint reviews reveals an underlying value of person focus (V4) through an emphasis on team empowerment.
“Something gone wrong and it becomes all hands on deck to fix it. It’s the ‘Limerick’ sprint review - not the code that Johnny wrote or Mary tested.”

Developer (QA)

The measurement of progress through working software (g13) was referred to by a developer who described the potential embarrassment of not meeting commitments at the sprint review. This fear of embarrassment is interpreted as a motivating factor underpinned by a value of individual empowerment (V4). In relation to the pursuit of sustainable development (g14), a respondent described how this practice was enabled by the management of stress through teamwork and collective responsibility. This emphasis on teamwork reveals a value of ‘person focus’.

“I think the stress level is less because it is spread among people. You have an opportunity to get help.”

Developer (SE)

8.6.5 ‘V5’ Continuous Improvement

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g1</th>
<th>Goal - g4</th>
<th>Goal - g6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Improvement</td>
<td>Visible feedback on productivity. Reviews to propagate learning to wider organizations. Apply rigorous standardization to establish agreed baselines for improvement. Actively seek and manage obstacles through agreed processes, such as root cause analysis. Promote organizational learning and a culture of relentless improvement. Use simple powerful tools.</td>
<td>Built-in instability</td>
<td>Multi-level learning</td>
<td>Effectively disseminate learning throughout Organization</td>
</tr>
</tbody>
</table>

Table 8-22: Candidate Value Rationale for V5 - ‘Continuous Improvement’

The majority of findings associated with this value referred to goals pursued within the post-sprint method fragment. Only one of the three candidate NPD goals were identified in the findings - g6. The additional goals to emerge were:

- Self-organization (g2)
- Best work emerges from self-organizing teams (g17)
- Regular reflection and performance improvement (g18)
- Alleviate GSD coordination challenges (g20)
The team changed their development approach to promote knowledge sharing in order to improve the quality of requirements. This activity is interpreted as the pursuit of self-organization (g2). This goal was pursued in order to improve their work, which reflects a value of continuous improvement (V5). This finding also indicates a value of waste removal (V2) due to the prevention of problems arising from poor comprehension of requirements).

“We found that our planning was not deep enough. We have added something the next day called an Acceptance Criteria meeting ... Planning is done when you have all your acceptance criteria.”

Developer (SE)

Pursuit of the goal of organizational transfer of learning (g6) is evident in the propagation of learning to the wider organization. The pursuit of this goal is underpinned by a value to continuously improve knowledge of development issues among the support team. It is observed through meetings and the sharing of artefacts between developers and other groups within the organization.

“We have an email group for each Scrum team and there are support people on those as well so when you use the group pdl (public distribution list) ... Anybody can join the review - there is nothing to hide”

Developer (QA)

A description of how modifications are made to the practice of performing retrospectives in the post-sprint fragment pursues the goal of self-organizing teams (g17). It is deemed more successful than traditional post-mortems due to immediacy of reflection and the ability to change an approach or process on the next working day. This practice achieved the goal of self-organization and is a simple powerful tool that enables relentless improvement and promotes reviews to seek and resolve obstacles.

26 SE: Developer whose core competency is as a software engineer. Name omitted to preserve anonymity.
“With retrospectives, we implement the things that we highlight for change...you see quickly if you are going wrong again. You go off the back of the retrospective and into planning so you are very conscious of things that did not go well in the previous sprint.”

Developer (QA)

Many reflections on the retrospective practice described it as the pursuit of the goal of regular reflection and improvement (g18). In such instances the value of continuous improvement is evident from use of an agreed process to seek out problems and act upon proposed solutions.

“We have done retrospective and we have applied the recommendations. Because of flowing from one sprint to another and also because of the same people”

Developer (SE)

From a GSD perspective, one respondent explained that he felt that the alleviation of coordination challenges through sharing of learning and feedback among the distributed team is not evident - but he did accept that this would be useful, thus indicating an underlying value of continuous improvement.

“Perhaps we should be sharing things that we find very good. We have not done that to date.”

Developer (BA)

### 8.6.6 ‘V6’ Product Excellence

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Excellence</td>
<td>Promote strong design culture to avoid premature convergence on incorrect solution. Prototype different options to derive best approach. Deliver functionality in change-tolerant form. Build quality in through practices of jidoka and poka-yoke. Promote deep specialized knowledge of product and process. Promote a culture of excellence.</td>
<td>Continuous attention to technical excellence and good design enhances agility.</td>
</tr>
</tbody>
</table>

Table 8-23: Candidate Value Rationale for V6 - ‘Product Excellence’

Although two single instances of other goals being underpinned by this value were identified, all other instances reflected a strong association between the candidate goal, technical excellence (g15) with product excellence (V6). The two additional goals identified were:
• Regular collaboration between developers and business people (g10)
• Best work emerges from self-organizing teams (g17)

The majority of findings referred to both sprint and post-sprint practices that pursue technical excellence (g15). The post-sprint comments referred to activities in the review practice where architects would interrogate the team in order to focus on design considerations. It seems reasonable to interpret that such focus is underpinned by a value of product excellence. Within the sprint, various reflections on technical activities (CS-8) such as design optimization and effective quality assurance practices indicate an underlying value of product excellence. Some reflections reveal instances where pursuit of the goal, while desirable, is not always achieved. One developer described how the lack of automated tests is thwarting the achievement of continuous attention to technical excellence (g15). Such concerns reveal an underlying value of product excellence.

“We are trying to do automated test but because of our legacy code it makes it harder - there can be a few drawbacks on the kind of technology we would like to use”

Developer (SE)

Another reflection explains how the complexities and restrictive processes used with the internal development framework act as obstacles to the pursuit of design optimization. However, the respondent’s comments reveal evidence of the underlying value of product excellence - he decries the lack of engagement in design issues.

“We are using this framework ... there is no design patterns. There is behind it but there is not much design going on to produce the solution. I have not seen a UML diagram...There is no OO design”

Developer (SE)

The point was raised that performing technical activities that achieved the pursuit of simplicity (g16) contradicted the goal of technical excellence (g15). Again this reflection indicates that although the goal of technical excellence may not have been achieved, its pursuit was motivated by a concern to promote strong design culture in order to avoid premature convergence on an incorrect solution which is one of the core principles of Product excellence (V6).
“You design something and put it in and then a few sprints later you notice that this could have been designed better if you had known about this earlier - this might be a drawback with the approach”

Developer (SE)

Another example of this value (V6) being present to promote a strong design culture is seen in the practice of product backlog grooming. The product owner pursues a goal to work daily with developers (g10) during the last week of the sprint in order to perform this practice and gain a more complete understanding of the requirements proposed for the forthcoming sprint. This is described by the business analyst in his answer to a question on whether he works closely with the other developers to determine the contents of the product backlog.

“I would not personally but the PO does. He uses them quite a bit as he is working in that last week to prepare for the following week”

Developer (BA)

Finally, reflections on the modification of the development approach to improve comprehension of the work being undertaken indicated the pursuit to be a self-organizing team (g17). This continual improvement to work approaches indicates an underlying value of product excellence (V6).

“Our planning was not deep enough. We have added something the next day called an Acceptance Criteria meeting. All the participants for a task delve into a story...”

Developer (SE)

8.6.7 ‘V7’ Embrace Change

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g1</th>
<th>Goal - g8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embrace Change</td>
<td>Defer commitment to scope. Encapsulate features and consider all options carefully. Facilitate emerging requirements.</td>
<td>Built-in instability</td>
<td>Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.</td>
</tr>
</tbody>
</table>

Table 8-24: Candidate Value Rationale for V7 - ‘Embrace Change’
The presence of this value in the pursuit of both candidate goals (g1 and g8) was evident in the findings. The pursuit of one additional goal was observed to be influenced by this value:

- Working software is the primary measure of progress (g13)

The NPD goal of built-in instability (g1) is pursued in sprint planning. Setting challenging targets for the team leads to instability that drives innovation and creativity, urging them to consider their options carefully and encapsulate required features. This supported the product owner to effectively prioritize sprint work. This reveals an underlying value to embrace change (V7).

“I usually oversubscribe so that folks don’t run out of work. They define the user stories and then I get to set the priority of the user stories.”

Product Owner

Whereas the above reflection is from the creator of the challenging targets, it is interesting to note a similar value of welcoming change is also evident from the receivers of those targets i.e. developers.

“I think it’s challenging but not unreasonably challenging. I think a challenge - goes with the trade. That goes back to change - you are implementing change and change is always challenging.”

Developer (BA)

The influence of embracing change (V7) on the pursuit of welcoming changing requirements (g8) is evident in many observations made by team members. In one instance, the Scrum process of presenting a new set of requirements at the outset of each sprint made the developer oblivious to the fact that he welcomed changes. Emerging requirements were facilitated as part of the normal process.

“You are doing work in small parts each time so it’s not a big thing to change...It’s kind of oblivious - the PO presents requirements at the start of each sprint and prioritizes as necessary.”

Developer (SE)
“The quick iterations is very good. You are not concerned whether the epics at the start of each sprint are related or not to previous work.”

Developer (SE)

A reflection on sprint planning described how the practice of definition of done (DOD) (CS-12) is associated with the goal of using working software as the primary measure of progress (g13) by clearly encapsulating the features to be delivered. Such careful consideration of options and feature encapsulation reveals an underlying value to embrace change (V7).

“Generally, this will outline the definition of done. We used to send it out for review but if everybody is in the meeting and we are all agreed we add it to Greenhopper. This story is complete when....”

Developer (SE)

8.6.8 ‘V8’ Business Environment Awareness

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Environment Awareness</td>
<td>Be aware of why you are doing work. Ensure it adds value to your situation. Future-proof your approach to work against changing market conditions. Consider domain solutions rather than restricting product to one market sector. Don’t force lean approach where inappropriate to business needs. Don’t optimize locally to the detriment of other aspects of the business.</td>
<td>Business people and developers must work together daily throughout the project.</td>
</tr>
</tbody>
</table>

Table 8-25: Candidate Value Rationale for V8 - ‘Business Environment Awareness’

The findings reveal that this value is an influence on the candidate goal identified - regular collaboration between developers and business people (g10). One reflection also highlighted the pursuit of an additional goal that had been influenced by V8:

- customer satisfaction (g7)

That reflection explained how an internal customer was able to give feedback on the value of each piece of work as it was completed. The internal customer was aware of why the work was being done and could determine whether it added value. They were also in a position to future-proof a solution against changing market conditions.
“There has been very regular communication ...between our BA and one of the people over there. As we finish something ... they can go into our environment and play with it. That is good as you get feedback.”

Developer (QA)

However, the pursuit of regular meetings with business people (g10) is noted as the main goal underpinned by this value. One such comment describes how this goal was pursued enthusiastically by the team, revealing a value of wishing to be aware of why work is being done (V8).

“We are very conscious of the CustomerName - there were cultural differences ... We were very conscious that we needed to engage on a weekly basis ... The whole team would know that the CustomerName phone call was coming up and they would question me after.”

Developer (BA)

Another instance of the team’s pursuit of this goal (g10) is evident in discussions about the daily Scrum meeting. This reflection indicates that the product owner’s presence at this meeting was deemed to be valuable because he enabled developers to gain an awareness of the reasons that they were developing certain features. He also could provide clarification on the validity of their solutions. These points indicate an underlying value of V8.

“PO attendance at Scrum ensures ...developers can be aware of why they are doing work and can confirm that their approach is appropriate and adds value to the customer”

Developer (SE)

8.6.9 ‘V9’ Data-driven Decisions

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Driven Decisions</td>
<td>Impartial data collection to drive decision making and reduce cost of meetings and disagreements. Rigorous scientific approach to continuous improvement.</td>
<td>No goals Identified</td>
</tr>
</tbody>
</table>

Table 8-26: Candidate Value Rationale for V9 - ‘Data-driven Decisions’

27 CustomerName: This was inserted to preserve the anonymity of the customer
No clear-cut goal-value associations were identified through the findings. However, certain reflections indicated a possibility that this value was evident in the consideration or pursuit of certain goals:

- Self-organization (g2)
- Working software is the primary measure of progress (g13)
- Sustainable Development (g14)

The product owner stated that he would like to pursue the measurement of progress through working software (g13). He did not feel that any of the Scrum practices helped him to achieve this goal. Although the value of effective process (V10) is likely to underpin his pursuit of the goal, he identified more specific needs. He felt that the effective pursuit of this goal required impartial data collection and a rigorous scientific approach to continuous improvement. These considerations indicate a value of data driven decisions (V9).

“We need to start looking at standards and I don’t think we have enough stats on software we deliver and how it works. And that brings up a whole other issue on how you quantify software?”

Product Owner

This issue is also reflected upon by a software engineer who questions the term ‘working’. It appears that the goal to use working software as the primary measure of progress (g13) is linked tightly with the practice of definition of done (CS-12). The product owner had earlier decried the lack of metrics to pursue this goal, which opens the issue of linking metrics to definition of done. It appears that an underlying value here is data-driven decisions to support accurate measurement of working software. (V9)

“What is working software? ( in response to question about any practices or experiences that he performs in order to ensure that working software is the primary measure of progress)”

Developer (SE)

Another comment addressed the pursuit of sustainable development (g14). The use of a burn-down chart (CS-9) is intended to gather data in order to effectively measure progress. However,
the goal is not achieved due to spikes that lead to the introduction of new stories with associated
time costs. It is not permitted to re-baseline the projected time and thus presents an invalid
picture. If one assumes that the creation of an estimate is a decision, then the goal to promote
sustainable development using solid progress data is underpinned by data driven decisions (V9).
He felt that the burn-down chart had been reduced to an artificial tool that was massaged for the
benefit of the tool rather than the development situation. Such misuse of a tool to support method
recommendations without appropriate understanding of its usefulness to the development context
is cited as a symptom of poor method application (Introna and Whitley, 1997). Other evidence
that this goal is pursued is evident from the perception by the developer that management use
this practice to monitor progress.

“If you add a story after the first day, and you add time to it, it does not affect the burndown chart but it
will affect the time recorded line.. I think it’s purely for management ... We won’t add tasks because the
tool shows it incorrectly so you are achieving more but it’s not shown”
Developer (SE)

A goal of cross-fertilization or knowledge sharing drove the group to plan together and discover
more about the work. This aspect of self-organization (g2) is further emphasized in this example
as they modified their approach based upon definite evidence of issues. This is not as rigorous as
it could be but is very close to the value of ‘Data Driven Decisions’.

“Using story points...you learnt things that you did not know as a result of the discussion between low
and high. It is a discovery phase. We tried to make them only one story point ... This caused everything to
come up to test in the last week which did not work.”
Developer (SE)

In summary, the findings on this particular value were not very clear. A value of data-driven
decisions seemed to underpin a need for additional rigour in order to link planning and
estimation to progress measurement. Such measurement could be facilitated within the sprint by
rigorous task estimation and monitoring and also across sprints by rigorous specification of the
definition of done.
8.6.10 ‘V10’ Effective Process

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g13</th>
<th>Goal - g18</th>
<th>Goal - g21</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mechanisms for resource allocation including consistent roles, names, work</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>practices. Ability to assess team productivity rate. Central expert to integrate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>overall development (‘Chief Engineer’). Scalable approach. Clear product</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>development roadmaps. Fast powerful feedback loops to build momentum.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Working software is the primary measure of progress.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>At regular intervals, the team reflects on how to become more effective, then</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>tunes and adjusts its behaviour accordingly.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8-27: Candidate Value Rationale for V10 - ‘Effective Process’

Of the three candidate goals proposed to be underpinned by this value, evidence of g13 and g21 were found. However, the findings also revealed a much broader spectrum of goals that were influenced by this value:

- Built-in Instability (g1)
- Self-organization (g2)
- Subtle Control (g5)
- Delivery of working software frequently and preferably early (g9)
- Motivated and trusted developers given suitable environments (g11)
- Sustainable Development (g14)
- Alleviate GSD coordination challenges (g20)

An overarching practice within the general considerations method fragment is to configure the team in a Scrum formation. The performance of this practice enables the team to self-organize (g2). The following excerpt describes how the QA is able to integrate into this formation while maintaining her other responsibilities. She took a full part in the pre and post work and used 50% of her work day to engage in sprint activities. This arrangement supported cross-fertilization within the team. Due to consistent roles and clear mechanisms for resource allocation, the QA was empowered to self-organize effectively so this goal was underpinned by the value of effective process (V10).
"I am 50% assigned to the Scrum team - what it mostly means is that I do attend the pre and post script activities whereas during the Scrum I would not be 100% assigned."

Developer (QA)

Reflections on the pre-sprint activities reveal that a consistent story pointing technique helped to achieve the goal of sustainable development (g14). Such a goal relies on clear comprehension of the method-in-action and the ability to assess the team productivity rate. These characteristics relate directly to the principles outlined above in the description of the value of effective process.

"We are getting there (achieving sustainable development). The pre-sprint - we don’t have very big stories any more. Maybe a 5 point story and we have a good idea of what you can pick off."

Developer (SE)

Another developer’s negative reaction to the story point estimation technique also revealed an underlying value of V10. At one point in the interview with this individual, he stated that the story point estimation technique impacted morale thus contradicting the goal of team motivation (g11) as it does not appear to support the work. However, he later described the same technique as being effective in the determination of team velocity. It appears that story point application may work but the reason for this was a mystery to him and as such it had a negative impact. The value underlying this contradiction appears to be effective process (V10). The developer wished to work with an approach that enabled clear comprehension of the method-in-action.

"If you are doing similar work all the time you can associate a number with work without relying on time. But over the last while we have been doing very distinct and different chunks of work. I'm not sure how relevant these story points are”

Developer (BA)

In relation to the alleviation of GSD challenges, two goals emerged from reflections on pre-sprint activities. Management of the product backlog alleviated coordination complexity between distributed teams (g20). The management of clear product development roadmaps indicates that the influential value in this case is effective process.

"We have 4 POs - one for each team... that gives the broad approach. We agree what each team is going to do .... That is done at the start of a new piece of functionality (not a sprint)."

Product Owner
The use of clear roadmaps is seen in another instance to indicate the influence of V10 on the pursuit of addressing GSD control challenges (g21). Building a clear release backlog alleviated challenges such as requirements changes between distributed sites.

“We have a roadmap defined for our full year. So we know whether something might be off the roadmap”

Product Owner

The business analyst reflected on a customer meeting practice performed during the sprint in order to deliver working software frequently (g9). Although this meeting referred to the post-sprint practice of demonstration and review, it was conducted during the next sprint. It reported to the customer on the demonstration contents and their impact on the overall release. In this instance, this goal is underpinned by a value of effective process (V10) as the team sought to receive information on their work through fast powerful feedback loops.

“As soon as the demo was over and on the Monday after the acceptance criteria meeting I would show them the demo that we had presented the previous Friday.”

Developer (BA)

The influence of this value on the pursuit of subtle control (g5) is evident in reflections on post-sprint activities. The demonstration gave the team freedom to determine how to perform their work. This freedom was confined to the boundary imposed by the three week sprint and the demonstration facilitated management control. This reflected a value of effective process (V10) as this subtle control enabled fast powerful feedback and presented an opportunity to assess the team’s productivity.

“New functionality you have added - show it to me. At the minimum it needs to be working.”

Developer (BA)

Another aspect of the influence of this value on subtle control (g5) is evident through work allocation practices. The product owner permitted the team to determine how much work was involved in items but tempered this non-invasive management with the right to re-prioritize the
items in order to get a commitment on the completion of particular pieces of work. This use of a central expert to integrate the overall development through work prioritization revealed an underlying value of effective process.

“We ... explain that we won’t get all the work done after breaking down the requirement... The PO accepts this ... he prioritizes the user stories to see what has to be done for this sprint”

Developer (SE)

Pursuit of the goal of progress measurement through working software (g13) resulted in a review practice that produced regular feedback. The establishment of such fast powerful feedback loops indicates an underlying value of effective process.

“At the end of the three weeks, all tasks completed are demonstrated. Incomplete tasks are explained and put in the backlog for future sprints”

Developer (SE)

Another finding from the sprint review showed that the pursuit of in-built instability (g1) is influence by effective process. In this instance, there is acknowledgement that reviews do not always go well but it is accepted that these can be good reviews if there is learning adopted by the teams to address why their targets are not met. This indicates the appreciation for fast powerful feedback loops to enable change.

“In our last sprint we found something that should have been tested that wasn’t. But I think that is still a good review because we discovered something that should have been done.”

Developer (SE)
8.6.11 ‘V11’ Effective Collaboration

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g3</th>
<th>Goal - g12</th>
<th>Goal - g19</th>
<th>Goal - g20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Collaboration</td>
<td>Reduce distance between collaborators and processes. Simple visual measures of productivity and quality. Balance functional expertise with cross-functional integration. Promote teamwork. Integrate suppliers. Unambiguous communication procedures.</td>
<td>Effectively Overlap development phases</td>
<td>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</td>
<td>Overcome GSD communic. challenges</td>
<td>Overcome GSD coord. challenges</td>
</tr>
</tbody>
</table>

Table 8-28: Candidate Value Rationale for V11 - ‘Effective Collaboration’

All four candidate goals were deemed to be present in the findings. Certain additional goals were also seen to be influenced by V11:

- Built-in Instability (g1)
- Self-organization (g2)
- Multi-level Learning (g4)
- Organizational transfer of learning (g6)
- Simplicity (g16)
- Alleviate GSD control challenges (g21)

There were more findings associated with this value than any other of the twelve LSD values. However it should be noted that of all the findings gathered in relation to this value, three goals dominated the results - g12, g19 and g20.

A consistent theme of teamwork promotion underlies many of the goals pursued. The pursuit of In-built Instability (g1) is intended to provoke a tension that promotes innovation and teamwork. This is evident from reflections on sprint planning work as the team address situations where story point allocations differ between team members.
“The person with the highest justifies theirs and the person with the lowest justifies theirs... sometimes people just concede... complexities that somebody might see may have not been anticipated by others”

The format of the daily Scrum aids in the pursuit of simplicity (g16) resulting in an effective meeting with clear rules. This appears to be underpinned by a value for effective collaboration. In this instance the daily Scrum rule of 3 questions had been augmented to address the value that technology must be used effectively. This required compliance to particular tasks such as logging time in order to ensure that the burn-down charts reflect up-to-date performance.

“Go around the room and ask the 4 questions: what did they do before the meeting, what will I do next, have you any obstacles and have you updated greenhopper (because it wasn’t being utilized properly)”

Developer (SE) - Scrum-master

Inviting the support and service personnel to the formal sprint review was motivated by organizational transfer of learning (g6). This was underpinned by a value to have effective collaboration and integrate other aspects of the organization. (One can argue that support personnel were suppliers to the product development team).

“We invite support people to our sprint review. The attendance isn’t great - but we have 1 or 2 each time. They may find it useful. Compared to pre-Scrum financials development, our handover is smaller”

Developer (SE)

In the following excerpt, overlapping development phases (g3) was achieved as testers stayed alert to the estimates being presented by developers. Pursuit of the goal of ‘shared division of labour’ reflects a value of effective collaboration by enabling the team to balance functional expertise with cross-functional integration.

“Sometimes I would not have thought of how much testing might be needed and QA 28 would speak up and highlight the need for more testing”

Developer (SE)

28 QA: Quality Assurance developer name omitted to preserve anonymity
Many reflections discussed the pursuit of effective communication (g12). The goal of conveying information face-to-face was deliberately pursued by structuring the overall team in a distributed Scrum of Scrums configuration. This refers to the reduction of distance between collaborators, indicating a value of effective collaboration. Also, this value appears to have been present in the collocated Limerick Scrum team’s decision to establish physical seating arrangements to support this goal.

“We are physically sitting beside one another. We even changed that a bit. We used to break into sub-teams. But we said we should all sit together to ensure that QA has visibility to what is going on”
Developer (QA)

Daily meetings and other side-meetings held in face-to-face settings are both efficient and effective in conveying information within the team (g12). One description of the power of such meetings conveys an underlying value to effectively collaborate.

“I’m probably worse than anybody spending a day or two banging your head off the screen whereas the daily Scrum helps avoid that. I can see how daily Scrum can bring people out of their shell.”
Developer (SE)

The value of effective collaboration is evident in the pursuit of all three GSD challenge alleviation goals. The perceived threat from low-cost alternatives (g21) was alleviated by the constant positive feedback and exposure received from whole team demos. Although it’s unlikely that such a goal was the primary motivation for the sprint review practice, the following excerpt indicates that achievement of the goal reduced cultural distance and supported effective collaboration (V11).

“Wages would be cheaper in XX²⁹ - that could be considered a threat. Unintentionally, Scrum may help to alleviate that threat. Each 3 weeks there is demonstrable s/ware that stands up.”
Developer (BA)

Working within the structure of 3 week sprints helped to alleviate the challenge of managing different frames of reference. Reducing communication challenges (g19) by having

²⁹ XX: Location of team omitted to preserve anonymity
unambiguous communication procedures supported by the focus and structure of a short iteration would appear to be influenced by the value of effective collaboration.

“That is true (that working in 3 week sprints helps distributed meetings). I am on the tech rep meeting and Scrum is helpful for people to discuss what is happening in those 3 weeks.”

Developer (SE)

Another practice designed to alleviate GSD challenges was the establishment of standard meetings. The specific challenge alleviated was difficulty initiating contact caused by increased geographical distance (g19). Again, effective collaboration is seen as the value that underpins the pursuit of this goal.

“We meet every Thursday with the product owners and if PO has brought anything back we can review and agree at that point. These meetings are held weekly for escalations, questions, etc.”

Product Owner

The majority of findings indicated that GSD coordination challenge alleviation (g20) was underpinned by a value of effective collaboration. The Scrum review practice enabled all of the teams to see what remote sites are producing and share feedback and knowledge.

“Somebody builds an API ... they created something that was not as usable as it should have been if they understood the bigger picture.... we probably caught it due to the reviews we had in Scrum”.

Developer (SE)

Many reflections by respondents were very positive toward the Sprint demonstration practice. However, there are also quite a few points made about how this practice may actually contradict the goal of establishing ‘teamness’ because sub-teams are so focussed on their own specific deliverables (cf. section 8.5.4). Such negative comments reflect an underlying value to effectively collaborate.

Although certain standard meetings addressed difficulties with initiating contact, there were also regular meetings organized to manage coordination complexity (g20). One example is the Scrum

---

30 PO: Product owner name omitted to preserve anonymity
of Scrum masters meeting which was performed in order to enhance coordination between the four distributed teams.

“I have a meeting on a Tue and Thu morning where I meet with the other Scrum-masters from each Scrum team... meet with SMOSM\(^{31}\). For 30 minutes we give our updates on any obstacles.”

Developer (SE) - Scrum Master

### 8.6.12 ‘V12’ Effective Use of Technology

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective use of technology</td>
<td>Technology must serve a particular need. Configuration management systems and quality measurement devices are examples of key facilities. Pursue autonomation by using technology to enable workers to effectively adjust system when necessary.</td>
<td>No goals Identified</td>
</tr>
</tbody>
</table>

Table 8-29: Candidate Value Rationale for V12 - ‘Effective Use of Technology’

Although no candidate goals were associated with this value, the pursuit of a number of goals by practices within the Sprint method fragment were interpreted to have been underpinned by effective use of technology (V12):

- Motivated and trusted developers given suitable environments (g11)
- Sustainable Development (g14)
- Simplicity (g16)
- Alleviate GSD coordination challenges (g20)
- Alleviate GSD control challenges (g21)

One of the first goal instances described in the previous section (8.6.11) noted that the pursuit of simplicity (g16) exhibited in the daily Scrum was underpinned by effective collaboration. However, another value was deemed to be present in this instance. While maintaining the simple meeting structure, the team augmented the daily Scrum rules to effectively use technology. A fourth question had to be addressed by each participant in order to verify that they had logged

\(^{31}\) SMOSM - Scrum-master of scrum-masters name is removed to preserve anonymity
hours worked in the task-tracking software application (JIRA). This would help to ensure that burn-down charts would effectively represent team performance.

“We have a daily Scrum meeting each day at 12:30... and have you updated greenhopper (because it wasn’t being utilized properly) That was one thing we added to the meeting”  
Developer (SE) - Scrum Master

Another reflection on the pursuit of simplicity explains that the lack of automated testing made it difficult for teams to restrict work to the bare minimum required as they were unable to in refactoring code in later sprints.

“Very hard to (keep it simple) at early sprints. We’re getting better but we don’t have automated testing so we are afraid to change code from earlier sprints. We don’t do refactoring because of all the testing.”  
Developer (SE)

A number of respondents reflected upon how the goal of providing needed environments (g11) was being pursued but not achieved. Such reflections are underpinned by a value to pursue autonomation by using technology to enable workers to effectively adjust systems when necessary (V12).

“Here the difficulty for unit tests is that you have to build a big environment before creating a simple unit test. They are trying to see how to fix this.....”  
Developer (SE)

It should be pointed out that some effort had been made by the team to create automated tests. A different instance explained how autonomation was being pursued and achieved in order to enable sustainable development (g14) to support the code merge process.

“We have automated tests to help this (the merge process)...We are looking at using different tools ”  
Developer (QA)

Technology was used to help overcome GSD coordination challenges (g20). Manual processes have been replaced by technology solutions in order to pursue this goal. The technology solution
served a particular need and as such, this goal reveals an underlying value of effective use of technology.

“Initially using post-it notes. Then greenhopper. Started off using the board. But we need the automated tool to give visibility to distributed teams.”

Developer (QA)

A similar example of this value influencing GSD alleviation challenges is evident in the experience of the Limerick Scrum-Master’s involvement in Scrum of Scrum Masters meetings. The burn-down chart was used to alleviate the challenge of controlling a project across distributed teams (g21).

“SMOSM\textsuperscript{32} would be looking at the chart a lot and seeing how our work is going.....also there is another product owner meetings with the higher management and they look at the burndown chart then as well”

Developer (SE) - Scrum Master

8.6.13 Summary of LSD Findings

Table 8-30 presents an overview of how responses related to the team’s working practices have been interpreted in order to determine the refined value rationale associating goals pursued by individuals with the LSD values that underpin them. All responses were examined in order to identify any LSD principles evident from the language describing the goal pursued by each practice. This examination reviewed both the raw excerpt provided by the respondent and the memo describing the analyst interpretation of the goal rationale. In certain cases, two LSD values were uncovered. In such cases, one value was considered to be dominant but the other value was still recorded and reported in the table surrounded by parentheses. For example, one particular finding showed that pursuit of the goal ‘G13’ was underpinned by ‘Waste reduction’ - V2 and to a lesser extent, ‘Flow of Value’ - V3. In that case, the V2 row reports the goal as normal, whereas in the V3 row, it is reported in parentheses under sprint findings. The total number of secondary goals is listed in parentheses and each goal code is also listed. In that example, the sprint goals column for V3 shows that G13 is the secondary goal in question by listing it as ‘(g13)’.

\textsuperscript{32} SMOSM - Scrum-master of scrum-masters name is removed to preserve anonymity
All the goals associated with a particular LSD value are listed in the ‘All Goals’ column. Further detail on the number and particular goal codes associated within each method fragment are provided in the additional columns on each LSD row. In certain instances, the pursuit of some goals did not seem to have been fully underpinned by the value. In such cases, the number of goals is suffixed by an asterisk and in the list of goal codes, any such goals are marked with an asterisk.

A few responses indicated that the pursuit of a particular goal appeared to exhibit a contradictory value to one of the LSD values. These instances are marked by a suffix of two exclamation marks (!!) on the number of goals affected and also as a superscript on the particular goal codes.

Effective collaboration (V11) is seen to be the LSD value most prevalent among the findings. Sixty seven findings were underpinned by this value. The majority of these findings were related to goals pursued while performing sprint and post-sprint practices.

The other most widely associated value was person focus (V4). Thirty eight responses described the pursuit of different practices that appeared to be underpinned by a value of teamwork, multi-skilling and individual empowerment.
<table>
<thead>
<tr>
<th>LDScrum Code</th>
<th>General Code Description</th>
<th>All Goals</th>
<th>Gen</th>
<th>Gen Goals</th>
<th>Pre-sprint</th>
<th>Pre-Sprint Goals</th>
<th>Sprint</th>
<th>Sprint Goals</th>
<th>Post-sprint</th>
<th>Post-sprint Goals</th>
<th>General Code Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Customer value</td>
<td>g7, g9, g10, g11</td>
<td>3</td>
<td>g7</td>
<td>7</td>
<td>g7, g7, g11</td>
<td>1</td>
<td>g10</td>
<td>3</td>
<td>g7, g9</td>
<td>Ensure customer defines value. Seek out user needs, not just requirements. Early delivery of value. Enable future needs (maintainability). Enable efficient deployment of product features. Provide value for money.</td>
</tr>
<tr>
<td>V2</td>
<td>Remove waste</td>
<td>g2, (g2), g5, g9, g13, g14, (g14), g15, g16</td>
<td>11(2)</td>
<td>g16, (g2), (g14)</td>
<td>6</td>
<td>g5, g13, g14, g14, g15, g16</td>
<td>5</td>
<td>g2, g9, g13</td>
<td></td>
<td></td>
<td>Promote systematic defect prevention. Eliminate rework through emphasis on customer needs. Holistic focus on quality that motivates the refactoring of legacy code to improve general product. Identify silos of waste present in value stream and improve processes. Promote reusability where appropriate. Take cognisance of the fact that waste can have different forms in different projects/work situations.</td>
</tr>
<tr>
<td>V3</td>
<td>Flow of value</td>
<td>g2, g8, g9, g13, g14, g16</td>
<td>3</td>
<td>g9, g13, g16</td>
<td>13</td>
<td>g2, g8, g13, g14, g16</td>
<td>6(1)</td>
<td>(g13), g8, g13, g16</td>
<td></td>
<td></td>
<td>Stabilize development process to enable levelled flow of value - manage necessary variability. Smooth demand from users. Pull value from user demands through lifecycle. Apply minimalism (scope items, teams, documents). Allocate resources only when needed.</td>
</tr>
<tr>
<td>V4</td>
<td>Person focus</td>
<td>g1, g2, g3, g4, g5, g11, g13, g14</td>
<td>5</td>
<td>g2, g3, g5</td>
<td>11</td>
<td>g1, g2, g3, g4, g5, g11</td>
<td>9</td>
<td>g3, g4, g5, g11, g13</td>
<td>13</td>
<td>g1, g4, g5, g11, g13</td>
<td>Empower individuals and teams. Promote continual worker development. Invest in multi-skilling and promote attitude to participate in multiple roles.</td>
</tr>
<tr>
<td>V5</td>
<td>Continuous Improvement</td>
<td>g2, g6, g17, g18, g20</td>
<td>1</td>
<td>g2</td>
<td>3</td>
<td>g6, g20</td>
<td>14*</td>
<td>g6, g6, g17, g18, g18, g20</td>
<td></td>
<td></td>
<td>Visible feedback on productivity. Reviews to propagate learning to wider organizations. Apply rigorous standardization to establish agreed baselines for improvement. Actively seek and manage obstacles through agreed processes, such as root cause analysis. Promote organizational learning and a culture of relentless improvement. Use simple powerful tools.</td>
</tr>
<tr>
<td>V6</td>
<td>Product excellence</td>
<td>g10, g15, g17</td>
<td>2*</td>
<td>g10', g17</td>
<td>9*</td>
<td>g15', g15</td>
<td>3</td>
<td>g15</td>
<td></td>
<td></td>
<td>Promote strong design culture to avoid premature convergence on incorrect solution. Prototype different options to derive best approach. Deliver functionality in change-tolerant form. Build quality in through practices of jidoka and poka-yoke. Promote deep specialized knowledge of product and process. Promote a culture of excellence.</td>
</tr>
<tr>
<td>V7</td>
<td>Embrace change</td>
<td>g1, g9, (g8), g9</td>
<td>1</td>
<td>g1</td>
<td>6(1)</td>
<td>g1, g8, (g8), g13</td>
<td>1</td>
<td>g8</td>
<td></td>
<td></td>
<td>Defer commitment to scope. Encapsulate features and consider all options carefully. Facilitate emerging requirements.</td>
</tr>
<tr>
<td>LDScrum Code</td>
<td>General Code Description</td>
<td>All Goals</td>
<td>Gen Goals</td>
<td>Pre-sprint</td>
<td>Pre-Sprint Goals</td>
<td>Sprint</td>
<td>Sprint Goals</td>
<td>Post-sprint</td>
<td>Post-sprint Goals</td>
<td>General Code Description</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>------------</td>
<td>-----------------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td>----------------------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>V8</td>
<td>Business environment awareness</td>
<td>g7,g10</td>
<td></td>
<td></td>
<td>6</td>
<td>g7, g10</td>
<td></td>
<td></td>
<td></td>
<td>Be aware of why you are doing work. Ensure it adds value to your situation. Future-proof your approach to work against changing market conditions. Consider domain solutions rather than restricting product to one market sector. Don’t force lean approach where inappropriate to business needs. Don’t optimize locally to the detriment of other aspects of the business.</td>
<td>6</td>
</tr>
<tr>
<td>V9</td>
<td>Data driven decisions</td>
<td>g2,g13,g14</td>
<td>2</td>
<td>g13</td>
<td>1*</td>
<td>g2*</td>
<td>1</td>
<td>g14</td>
<td></td>
<td>Impartial data collection to drive decision making and reduce cost of meetings and disagreements. Rigorous scientific approach to continuous improvement.</td>
<td>4</td>
</tr>
<tr>
<td>V10</td>
<td>Effective process</td>
<td>g2,g5, g9,g11, (g11), g13, g14, g20,g21</td>
<td>1</td>
<td>g2</td>
<td>5</td>
<td>g5,g11,g14,g20,g21</td>
<td>2</td>
<td>g9, g20*</td>
<td>4* (1)</td>
<td>g1, g5, g9, g13, g14, (g11)</td>
<td>Easily understood method. Enable clear comprehension of method-in-action. Clear mechanisms for resource allocation including consistent roles, names, work practices. Ability to assess team productivity rate. Central expert to integrate overall development (‘Chief Engineer’). Scalable approach. Clear product development roadmaps. Fast powerful feedback loops to build momentum.</td>
</tr>
<tr>
<td>V11</td>
<td>Effective collaboration</td>
<td>g1, g2, g3,g12, g20,g21</td>
<td>9*</td>
<td>g12, g20*</td>
<td>10</td>
<td>g2,g3,g12, g20,g21</td>
<td>31*</td>
<td>g2, g3, g12, g19, g20, g20*, g21, g21*, g16</td>
<td>g4, g6, g12, g20, g20*, g21, g21*</td>
<td>Reduce distance between collaborators and processes. Simple visual measures of productivity and quality. Balance functional expertise with cross-functional integration. Promote teamwork. Integrate suppliers. Unambiguous communication procedures.</td>
<td>67</td>
</tr>
<tr>
<td>V12</td>
<td>Effective use of technology</td>
<td>g11, g14, g16, g20, g21 (g16)</td>
<td>8 (1)</td>
<td>g11, g14, g16, g20, g21 (g16)</td>
<td>83</td>
<td>59</td>
<td>223</td>
<td>223</td>
<td>Technology must serve a particular need. Configuration management systems and quality measurement devices are examples of key facilities. Pursue automation by using technology to enable workers to effectively adjust system when necessary.</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

*Note - * means that value not fully underpinning goal. !! Means that value is contradicted by this goal. (xx) means that this goal is secondarily underpinned by value and has been counted already in a primary value. Do not count (xx) in totals as they are secondary associations.

Table 8-30: Overview of all LSD findings
8.7 Refined LDScrum Model (Scrum of Scrums)

This chapter presented the findings from analysis of the use of the Scrum framework by the Fin-Limk team. This team worked as part of a distributed team using a ‘Scrum of Scrums’ structure. In order to support future related research activities, the chapter opened by augmenting the overall case study context presented in section 7.7 with a more detailed description of the Fin-Limk team and their project.

Chapter 5 explains how the LDScrum model was constructed from a combination of the aforementioned LSD Value rationale and the DScrum model. The DScrum model is a candidate Goal Rationale that links core Scrum practices to a set of goals that are proposed to motivate the application of those practices. Core Scrum practices are derived from literature analysis and are described in chapter 3. The goals are sourced from a combination of new product development characteristics (NPD) identified by Takeuchi and Nonaka (1986), principles of the agile manifesto (ASD) used by Ågerfalk (2006) and Global Software Development challenge alleviation goals (GSD) adapted from Ågerfalk et al. (2005). Chapter 3 also describes the NPD and ASD goals and associates them with the proposed core Scrum practices. This work is then extended in chapter 4 to describe the GSD goals and also associate them with core Scrum practices. This set of associations is amalgamated into a candidate distributed Scrum model (DScrum) in chapter 5 which uses method rationale analysis theory to capture the goal rationale linking core practices to the NPD, ASD and GSD goals. Finally, the LSD values are associated with the DScrum goals in order to construct a candidate value rationale. The combination of the goal and value rationales result in a conceptual model linking Scrum practices to goals and those goals to values. This model is labelled the candidate ‘LDScrum’ model.

The candidate DScrum model was used to support data collection and analysis. Findings were presented from the perspective of how each of the three different sets of goals may have influenced the performance of Scrum practices (‘refined goal rationale’). The findings were described and then summarized to show any refinements between actual practice-goal associations and the original candidate associations in the DScrum model. A fourth level of analysis reviewed the data to determine what LSD values may have underpinned the pursuit of
the aforementioned goals (‘refined value rationale’). These findings were described and compared to the initial candidate value rationale for each value.

Two summary tables capture the differences that emerged between the candidate and refined goal and value rationales as a result of this empirical analysis. Table 8-31 on the next page presents a refined version of the goal rationale arising from the findings related to the work of the Fin-Limk Scrum of Scrums team. Any boxes that contain the letter ‘c’ indicate practice-goal associations proposed in the candidate goal-rationale. On the following page, table 8-32 reflects the refined value rationale arising from findings that have emerged from the FIN-NN project. This table can be compared with the candidate value rationale proposed in chapter 5 and presented in table 8-17 at the outset of the previous section. The black boxes denote an association where a goal (represented by the row) has been underpinned by a particular LSD value represented on the column. As with the previous table, any black boxes containing the letter ‘c’ indicate goal-value relationships that had been proposed in the candidate LDScrum value rationale and were interpreted to be also present in the manner in which the Scrum of Scrum teams performed their work.
<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Goal</th>
<th>New Product Development</th>
<th>ASD Principles</th>
<th>GSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8-31: Refined Goal Rationale (Scrum of Scrums findings)
<table>
<thead>
<tr>
<th>LDScrum Goal</th>
<th>Code</th>
<th>Description</th>
<th>LDScrum Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NPD-1</td>
<td>Built-in instability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NPD-2</td>
<td>Self-organizing project teams</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NPD-3</td>
<td>Effectively Overlap development phases</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NPD-4</td>
<td>Multi-level learning</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NPD-5</td>
<td>Leverage Subtle control over work tasks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NPD-6</td>
<td>Effectively disseminate learning throughout Organization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-1</td>
<td>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-3</td>
<td>Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-2</td>
<td>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-4</td>
<td>Business people and developers must work together daily throughout the project.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-5</td>
<td>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-6</td>
<td>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-7</td>
<td>Working software is the primary measure of progress.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-8</td>
<td>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-9</td>
<td>Continuous attention to technical excellence and good design enhances agility.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-10</td>
<td>Simplicity – the art of maximizing the amount of work not done – is essential.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-11</td>
<td>The best architectures, requirements, and designs emerge from self-organizing teams.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A-12</td>
<td>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GSD-1</td>
<td>Overcome GSD communication challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GSD-2</td>
<td>Overcome GSD coordination challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GSD-3</td>
<td>Overcome GSD control challenges</td>
<td></td>
</tr>
</tbody>
</table>

Table 8-32: Refined LSD Value Rationale (Scrum of Scrums findings)
Chapter 9- Totally Integrated Scrum Findings

9.1 Introduction

As in the previous chapter, figure 9-1 presents the portion of work from the overall WBS (Figure 1-7) that is performed in order to present the findings for a particular project. Although work package 2.2 does not explicitly state any reporting activities, it is implicit that the analysis and establishment of value and goal rationales incorporates the reporting of all findings.

Chapter 9 presents findings related to another I-E-S project that is performed by a distributed team using the Scrum framework. The distinction between this research subject and the FIN-NN project described in chapter 8 is that this project is performed by a team that use a different distribution configuration: ‘Totally Integrated Scrum’ (cf. 4.3.5). This chapter uses the same structure as chapter 8. The context of the situation is described using the four ‘inner level’
contextual facets. This is followed by a description of the findings associated with the goal rationales for NPD, ASD and GSD goals. Each of these goal rationale descriptions is further decomposed into four Scrum method fragments: general considerations, pre-sprint, sprint and post-sprint. The LSD value rationale is then presented. The chapter concludes with a summary of the refined goal and value rationales that have emerged from this aspect of the research.

Decomposition of findings into method fragments within specific goal rationales was performed in order to use a rigorous analysis structure and to facilitate the construction of the refined LDScrum model. Another useful outcome of this approach is that due to the consistent presentation of findings in chapters 8 and 9, detailed descriptions of particular fragments may be compared. Chapter 10 presents the refined LDScrum model and explains how findings from both I-E-S projects were combined. Although this explanation discusses similarities and differences between the two projects, the reader may still wish to access more details on a specific aspect of how Scrum was used by the different project teams. The consistent presentation of fragments and values in chapters 8 and 9 should facilitate this need.

9.2 QUAL-NN Project: Totally Integrated Scrum Context

The QUAL-NN project is responsible for quality assurance across projects within I-E-S. This project team (QUAL) leverage a ‘management by projects’ approach to address a variety of broad quality issues (PMI, 2008). In essence, their work is driven by a series of ‘operational’ tickets that are organized into projects. Quality issues fall within three clear areas: (i) Systems Test (ii) Compliance and (iii) Quality Drive Initiative (QDI). Systems test work is primarily concerned with the maintenance and execution of a series of automated tests to inspect the quality of the integrated final product prior to release. These tests need to be executed prior to two major product release points each year. Compliance work relates to initiatives such as internal quality audits against various project teams worldwide in order to assure that their activities meet their commitments with respect to ISO certification requirements. Quality Drive
Initiative includes liaison with specific project teams in order to build and maintain automated regression tests for the particular software products under development.

An element of fragmentation is evident within the team as each of these three areas is self-contained and certain team-members are considered to be primarily responsible for particular areas. However, the team is moving toward a shared resource model and is using the Scrum framework to support this goal. The Quality Assurance line manager acts as a product owner. He is supported in this role by the leader of each of the three areas described above. The activities of these leaders would characterise them as ‘sub product owners’ as they are well-positioned to liaise with external stakeholders and also aid with prioritization of tasks within their specific area. Team members exhibit a mixture of business analysis, software engineering and quality assurance skills. The distribution configuration used is ‘Totally Integrated Scrum’. The product owner, Scrum-master and three other developers are based in Limerick, Ireland. Four other developers work from the corporate headquarters in California. The three QDI developers are based in Limerick, Ireland. Compliance work tends to fall under the remit of two developers in California. System test activities are currently the most widely shared tasks within the team. Although three developers (the Limerick Scrum-master and two California developers) are dedicated to this work, there has been notable success in promoting involvement in this work within the overall team. Many of the team members commented on the effectiveness of using Scrum when working with others to achieve a common goal (such as the successful management and completion of a diverse suite of system tests). The team has found that rigorously following Scrum practices has led to an increase in productivity.

“The team used to be much larger and the system test took 7 weeks. Now with a much smaller team, we get this done in 3 weeks.”

QUAL Product Owner

The QUAL team has traditionally engaged with all of the development teams as it supports system wide quality. It was inexperienced in the application of Scrum and was using the framework to support increased collaboration and coordination within the team. Because the QUAL distributed team used the Totally Integrated Scrum configuration, it was important to
analyse the views of all team members in order to get a rich picture of their work and collaborative activities. It should be noted that the use of this configuration by an immature Scrum team runs contrary to the recommendations of the Scrum founders (cf. section 4.3.5).

At the time that this Scrum team was formed, the most integrated activity being performed was system test. It appears that this may have been the motivation for the selection of the system test lead to play the Scrum-master role. This decision seems to have worked well as she has been in a position to facilitate coordination activities between the remote locations in order to streamline the system test tasks. The primary skill set of all three sub-PO roles was QA. At the time of writing, one of the California-based developers was being mentored into the QA role as their core competency was business analysis in the area of compliance (FDA regulations).

For the reasons outlined in chapter 7, the two I-E-S context facets presented in section 7.7.1 are augmented below with a description of the QUAL-NN project. Four ‘inner-level’ context facets are used to structure this description (Petersen and Wohlin, 2009).

### 9.2.1 Product

The I-E-S flagship product provides enterprise resource planning functionality that integrates manufacturing, logistics and financial functions. These core functions may be customized through extension points and ‘bolt-on’ products providing specialized logistics functionality to customers in the manufacturing sector. A key aspect of the development of this product and its associated add-on features is the application of quality activities that encompass quality engineering and integrated automated testing.

### 9.2.2 Processes

As stated in the previous chapter, the overall R&D organization of I-E-S conducts two release activities per year. Both the interim release in February and the major product release in September require the execution of a broad set of system tests in order to assure that the integration of many individual development projects has not damaged overall system quality.
Such regression testing is enabled within QUAL-NN through a series of automated tests maintained and executed by the QUAL system test group. Each of the individual I-E-S development projects that produce the sub-products and features that comprise the release are also supported by the QUAL compliance and QDI teams. Compliance activities support internal audits and quality engineering activities within the development teams. The QDI group assist quality assurance specialists within the development teams through the automation of their manual tests in order to support regression testing within their particular environments.

9.2.3 Practices, Tools, Techniques

Team members at each location worked in close proximity (low-partition cubicles). The team worked in three-week sprints. On the last day of the sprint (Friday of week 3), a meeting was held that combined the review and retrospective of that sprint with a sprint planning meeting to prepare for the sprint due to commence on the following Monday. This particular meeting was conducted during the California morning and Irish afternoon. The sprint backlog was formed from a series of job tickets that had been entered by the product owner and each of the three sub-product owners. These tickets were entered into the Atlassian JIRA issue-tracking system that was used by the wider I-E-S organization. Some tickets may have been entered by internal customers within the wider organization (such as support staff). However, in order for such tickets to become part of the sprint backlog, they would have been reviewed and approved by the sub-product owner associated with the subject matter addressed by the ticket. On the days preceding the sprint planning meeting, a form of ‘sprint backlog grooming’ was conducted by each sub-product owner. This involved both the selection of tickets to be completed in the sprint and also their assignment to particular developers. An estimate was also applied to each ticket. In many cases, these grooming activities were conducted with the particular developers involved in the domain area (such as QDI).

Prior to the sprint planning meeting, the team shared a form that enabled each developer to submit their available time during the next sprint. The sprint planning meeting leveraged AT&T teleconferencing software. Using this system, each developer shared a JIRA screen listing all sprint backlog tickets that were their responsibility. They described each ticket to the team and
there was an opportunity for discussion about the ticket and the proposed estimate. The product owner maintained a running count of all accepted estimates to ensure submitted capacity figures were not exceeded by the planned work.

During sprint execution, much of the work was performed on an individual basis. However, within each of the three sub-domains, developers will collaborate to overcome obstacles. A daily Scrum meeting was held to share knowledge of progress and highlight any obstacles that could impact each individual’s work. When a sprint involved a system test execution cycle, the daily Scrum was supplemented by a coordination meeting that ensured maximum team focus on achieving the combined goal of completing all planned tests across those particular tickets. This activity was acknowledged by many developers as being a particularly effective use of the Scrum framework, which was most likely due to the integrated team-oriented nature of this work.

The core activity of the final Friday of the sprint was the morning (USA)/afternoon (Ireland) meeting that commenced with the sprint review and retrospective and concluded with sprint planning for the next sprint. In order to prepare for this meeting, each individual developer ensured that all tickets assigned to them were documented correctly in JIRA and ready for review. The other main preparation required each developer to submit their views to a shared retrospective document. These views addressed the work practices used in the sprint and any potential improvements that could be implemented. The Sprint review did not involve any demonstration of completed software. Each developer shared the JIRA screens of their tickets and described the activities that led to their completion or alternatively, explained why a ticket has not been completed. It should be noted that the review meeting was restricted to the product owner and team and did not involve any other parties at that time. Following the review, a retrospective meeting used the shared retrospective document to facilitate a discussion on the work of the previous sprint. The goal was to elicit three themes for improvement from the observations of the group and establish corrective actions to implement these improvements in the forthcoming sprint.
A content management system (Atlassian Confluence) was used to manage documentation in relation to the system test, QDI and compliance work activities performed within the QUAL-NN project. A combination of these documents formed the product backlog. It was intended to increase access and visibility to these requirements using this tool as they have been fragmented to date. All requirements addressed by sprints were submitted as JIRA tickets. These were labelled and categorized according to their planned sprint backlog and as such, JIRA may be used as an historical repository of the outputs from each sprint.

9.2.4 People

As stated above there were eight people on the QUAL team and they were supported and managed by the product owner. All team members have been formally trained in the Scrum framework. Details of the primary roles of team members are presented in table 9-1. Most developers on the team had a background primarily in quality assurance. The amount of experience in the domain was varied and both the Limerick and California locations contained developers who were recent recruits and also developers who have worked in QUAL-NN activities prior to the adoption of the Scrum framework.

<table>
<thead>
<tr>
<th>Roles</th>
<th>Location</th>
<th>Limerick</th>
<th>California</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Scrum-Master</td>
<td>QA (She is also System test sub-PO)</td>
<td>1 (QDI) 1 (System Test)</td>
<td>1 (Compliance &amp; system test)</td>
</tr>
<tr>
<td>Developer (Sub-PO)</td>
<td></td>
<td>1 (QDI &amp; system test)</td>
<td>1 (System test)</td>
</tr>
<tr>
<td>Developer (SE)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developer (QA)</td>
<td></td>
<td>1 (QDI &amp; system test)</td>
<td>1 (System test); 1 Compliance &amp; system test)</td>
</tr>
</tbody>
</table>

Table 9-1: Configuration of 1-E-S R&D QUAL Scrum development team
9.3 Findings associated with New Product Development Characteristics

In order to present the investigation into the presence of NPD goals, the same approach is taken as outlined in chapter 8. A hypothesis statement is used as a controlling structure. In this section the hypothesis declaration is that all 6 NPD goals are present in a distributed Scrum team. The findings that emerged from research into the QUAL team activities are described using the method fragment structure used in chapter 8. However, quotes used to support findings indicate the location of the source as well as the role in this chapter. The reason for this distinction is to provide the reader with extra insight into the source of findings as all team members are not collocated which was the case for the Fin-Limk team. Each of the four sections below contrast the candidate goal rationale proposed in chapter 5 for that fragment with the empirical evidence interpreted from the research. To support this comparison, each section commences with a table that contains both the candidate goal rationale and refined (empirical) goal rationale for the method fragment addressed by the section. Each candidate practice-goal association is denoted by a tick mark (\( \checkmark \)). Each empirical practice-goal association is denoted by the letter ‘E’ (Empirical finding). Instances where a practice-goal association is present in both candidate and empirical results are shown by the presence of both the tick and the letter ‘E’ in the cell.

\[ \text{Table 9-2: NPD goals pursued by practices in ‘General Considerations’ fragment (adapted from table 3-11)} \]

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Scrum Practices</th>
<th>NPD Goals</th>
<th>New Product Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-1</td>
<td>Scrum Team Configuration</td>
<td>E</td>
<td>E √</td>
</tr>
<tr>
<td>CS-2</td>
<td>Initial planning (and ongoing) - create or update product backlog</td>
<td>√</td>
<td>E</td>
</tr>
<tr>
<td>CS-12</td>
<td>Definition of Done</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Initial analysis proposed that all of the NPD goals were pursued by practices within the general considerations fragment. There were responses that indicated pursuit of all goals except NPD-6.
The use of the Scrum framework and configuration (CS-1) to perform the system test component of the team’s work has led to a team that has responded positively to challenging targets. This was evident from the dramatic increase in productivity in this activity.

“It was 6 weeks to run the 300 + scripts. Using Scrum, it’s down to 2 weeks because everybody has their individual backlog document and the daily Scrum has greatly increased productivity”

Limk-Developer (QA & Sub-PO)

The increased productivity appears to be motivated by a greater sense of focus that has emerged from the built-in instability (NPD-1) provoked through use of the Scrum framework. Having freedom to determine how to address challenging targets appears to have resulted in a stronger awareness and tension around the need to achieve objectives.

“There is a lot more focus on the tasks that are in front of you and a larger awareness in the noise that is coming in from outside... All of us see that a lot more and that sort of makes it a lot more intense”

USA-Developer (QA & Sub-PO)

Aspects of self-organization (NPD-2) are evident in recollections of work practices and team configuration. The team initially had a standard approach of concluding sprints with a review and retrospective followed on the next working day by a sprint planning meeting to commence the next sprint. They determined that for their distributed team, it was more effective to manage the temporal distance challenge by combining all three meetings into one event. However, when there was a ten day gap between sprints for the Christmas holiday, the team reverted back to the original structure. This modification ensured that the post-holiday sprint would benefit from the momentum of a sprint-planning meeting occurring immediately before the commencement of sprint work. Another example of self-organization is evident from comments on cross-fertilization activities between team members leading to a stronger awareness of general quality activities.

“We have a larger visibility of quality across product. I feel the communication during system test is very good and valuable”

USA-Developer (QA & Sub-PO)
The achievement of collective responsibility through overlapping development phases (NPD-3) was noted by one respondent when commenting on how some team members are broadening their duties. This is remarkable due to the traditional individualistic nature of the work that is still evident from the practice of assigning work as tickets to individual team members.

“They would be chipping in on system test. Also potentially being trained up as auditors as well. There wouldn’t be really anybody who only does one role”

Limk-Developer (QA & Sub-PO)

In contrasting the general approach and structure of Scrum to their previous work approach, one respondent noted that Scrum had contributed to peer-learning and knowledge sharing (NPD-4).

“He tells me in the planning and he tells me what he did at the end of the sprint. Months ago we just had our one weekly meeting and I would not have that visibility”

Limk-Developer (QA & Sub-PO)

Development and management of a product backlog (CS-2) was motivated by subtle control (NPD-5). One respondent noted that a periodic meeting with the manager involved management of a large body of required work. Their description of how they address this work indicated that they have the freedom to determine what work should be addressed by their area within the team. They are acting as product owners for their own area although they do this within the control of the overall quality manager. There appears to be a product owner to sub-product owner relationship. This is borne through from evidence of the activities of the leaders of the other two areas within the team. There are three sub-product owners who each manage a large body of work and the combination of work from the three areas resembles a product backlog (CS-2).

“You would have a long list of work to do … I suppose in a way you are PO of your own area”

Limk-Developer (QA & Sub-PO)
Table 9-3: NPD goals pursued by practices in ‘Pre-Sprint’ method fragment (adapted from table 3-11)

Discussions around the creation of the sprint backlog (CS-4) showed that the team had to address challenging targets. This led to an increased emphasis on capacity management within sprint planning in order to try to ensure all the work was completed. Prior to the actual sprint-planning meeting, two of the sub-areas held meetings to determine the content of the sprint. These meetings afforded developers the opportunity to discuss their work in the sprint with the sub-product owner responsible for that area. These were unofficial meetings but appear to have been motivated by challenging targets and the fact that the team has the freedom to determine how best to meet these targets (NPD-1).

“We have added a meeting on our Wednesday before the planning meeting to have a discussion around compliance activity and be sure that we are in agreement on the sprint backlog”

USA-Developer (QA)

Prior to the planning meeting, the sub product owners would have entered JIRA tickets for themselves and the other developers working in their area. They also applied estimates to these tickets and the purpose of the planning meeting was to discuss the tickets and ensure that there was agreement on both the individual estimates and that the team had enough capacity to meet the work estimated for all tickets on the sprint backlog (CS-4). This approach to sprint planning appears to build in instability as challenging targets provoke debate. Much of this debate occurred in the unofficial meetings within the areas prior to the main sprint planning meeting but there is evidence that developers are empowered to review and modify work assigned to them where it was appropriate. A very extreme example of such a review was reported by one respondent in reference to estimates on technical work that had been planned and assigned to
him. In this case, assignment of the estimates had been aggressive and not considered certain aspects that the particular developer felt were necessary in order to complete the work.

“We were asked to review the estimates. Let’s say, I doubled all of them almost.”

USA-Developer (SE)

The above excerpt also reveals the level of autonomy given to developers in order to enable them to innovate on how best to meet challenging targets. This autonomy is seen to extend to the team and the evolution of the sprint-planning practice reveals a motivation to be a self-organizing team. Initially, the planning meeting was used to discuss the work to be done and developers were empowered to enter their estimates in tickets in order to manage sprint capacity. However this was not working and a decision was taken to have the estimates entered by the sub-product owners in order to facilitate effective discussion on the work to be done. A structure requiring each developer to present their tickets to all other team members in the sprint planning meeting (CS-4) shows a motivation to promote cross-fertilization. Many comments from respondents reflect that this goal has not been fully achieved due to the strong divisions that still remain between the different areas of responsibility within the team.

In the case of the systems test sub-area, numerous job tickets may be related. In such cases, the work was presented in the context of a user story in order to pursue collective responsibility (NPD-3). Each team member is made aware in the sprint planning meeting (CS-4) of how their tickets contribute to the achievement of an overall goal within the sprint (CS-5). Throughout the research, it was evident that because the requirements of this sub-area involve integrated tickets performed by multiple team members, it was seen as an area that has benefitted most from application of the Scrum framework.

“She may have a story addressing some system test cycle and make the statement that all these other people have their tasks that are associated with the story .... As we are cycling through all the people they are all talking about their own personal part of that story.”

USA-Developer (QA & Sub-PO)

There is little mention of multi-level learning (NPD-4) other than the observation by one of the sub-product owners that they continue to learn a lot about planning and task management due to
their continual exposure to the cycle of planning meetings at the outset of each sprint. The concept of subtle control (NPD-5) is evident in comments from both sub-product owners and other developers. Each sub-product owner had a strong knowledge of their sub-area and supported the product owner in prioritizing work to be done in each sprint. Developers used the sub-product owners to prioritize activities when it was clear that there was not enough capacity. All developers felt empowered to tackle their work in the most effective manner that they deemed appropriate.

“The planning meeting...we just discuss the tickets we created or the tickets that have been created for us and what we are doing with them, how to go about completing them”
USA-Developer (QA)

The sprint planning meeting (CS-4) sought to have each team member engaged and free to tackle their tasks within the boundaries of the sprint. This subtle control worked to an extent but other general enquiries on this process note that there was very little questioning of estimates at this meeting. This was due to the immaturity of the cross-fertilization thus far in the evolution of the team. As team members emerge from their exclusive focus on particular sub-areas or silos of work, there is a possibility that this process could be enriched through increased engagement by the wider team.

“Everybody would pick their own area in JIRA and would go through the task...for tickets I assigned to somebody else, I would ask them did they think the estimates were ok”
Limk-Developer (QA & Sub-PO)
9.3.3    Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Scrum Practices</th>
<th>New Product Development</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NPD-1</td>
</tr>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td>√</td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td>E √</td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td>√</td>
</tr>
</tbody>
</table>

Table 9-4: NPD goals pursued by practices in ‘Sprint’ fragment (adapted from table 3-11)

Although it had been identified as a candidate goal for sprint activities, built-in instability did not emerge from the research findings. Elements of cross-fertilization (NPD-2) were evident from the findings, especially in relation to the daily Scrum. However, most of the respondents found that the daily Scrum helped them to achieve collective responsibility as they sought to assist one another with ticket-related obstacles. As stated earlier, it is the system-test sub-area where this is found to be most significant. When the team was involved in system test execution, the daily Scrum was supplemented by a 15 minutes post-daily Scrum meeting just for team members engaged in this work so that further coordination and work-sharing may be planned.

“Especially in the daily meetings - somebody will announce that a certain test case is taking much longer than expected and somebody else upon hearing that will say that mine is doing fine and I can pick up the next one for you if you like”

USA-Developer (QA & Sub-PO)

“There is definitely overlap between all of us. The one area that I have no overlap into is the testcomplete area. But QA has. System test overlaps everybody”

USA-Developer (QA & Sub-PO)

“There is always 3-4 people executing tests within that backlog - this is where we can start to swarm a little bit and we each pick up activities from each other.”

USA-Developer (QA)

---

33 Candidate goal rationale: Each practice-goal association is denoted by a tick mark (√).
34 Refined goal rationale: Each practice-goal association is denoted by the letter ‘E’ (Empirical finding).
34 QA is a developer on the team. Their name is removed to preserve anonymity.
Collective responsibility (NPD-3) is also evident in the manner that certain team members performed general sprint technical activities (CS-8). In one sub-area, the team members sat in close proximity and would collaborate closely to overcome obstacles to their work.

“If I was stuck on a part of a test, we would definitely sit down together to figure it out.”
Limk-Developer (QA)

Another respondent considered this point from a GSD perspective by explaining that her colleague may be called upon by a remote team member to assist in overcoming a technical blockage.

“... works on another automation tool ... and she is trained up on the programming side of that. If the guy in the states has a problem, they would log in to the meeting room ... and work together to sort it out.”
USA-Developer (QA)

One team member was very clear in explaining that his role within the team has traditionally been very independent and that he has a lot of difficulty in seeing how the daily Scrum assists him in the performance of his job. He accepted that it was useful when asked to join with other team members in system test activities, but he felt that it was not useful for his primary work activities. However, upon reflection he posited that if cross-training was established and other team members were to become more proficient at certain technical activities, then the daily Scrum could help to promote collective responsibility in relation to the completion of his primary duties.

“If I am overwhelmed with GTA tickets and I know that QA1 and QA2 have some spare time, then they could help - it would be good to share the burden”
USA-Developer (SE)

Another practice that supported collective responsibility was management of the burn-down chart (CS-9). The burn-down chart was referred to at the end of the daily Scrum meeting each

35 QA1 and QA2 are two developers whose names have been removed to preserve anonymity
day and acted as an indicator for the reallocation of work within the team in order to correct any slippages occurring within the original sprint targets.

“We would look at the burn-down chart every day...and see where we are. SM\(^{36}\) ...might see that somebody has all their work done and somebody has not and she would automatically divvy it out.”

Limk-Developer (QA & Sub-PO)

Daily Scrum (CS-7) was also identified as a forum for sharing learning (NPD-4). One team member explained that if they have identified a workaround to an environment issue, the daily Scrum was the best place to broadcast this information to the team. Again, this practice was mentioned in the context of the system test work as tasks within this area are readily recognized and most relevant to the majority of team members.

The final goal identified as a motivator for Sprint practices is subtle control. The practice of locking features within a sprint (CS-6) was accepted as an aspiration by one respondent as unplanned work may have to be accepted in the middle of a sprint. A subtle control was established through the creation of a ticket to account for time required to address work that was unknown at the outset of the sprint. This could also be considered a planning mechanism to embrace change.

“There is unforeseen problems that we have to work on sometimes...PO\(^{37}\) created a ticket for whatever is not planned - we can enter hours against that ticket.”

USA-Developer (SE)

Another aspect of subtle control is evident in the approach taken to the prioritization of sprint activities. Developers were empowered to work on any tickets that were considered appropriate so long as they were on the sprint backlog. The developer was aware of customer needs and also could react to unforeseen obstacles such as unavailable work environments.

\(^{36}\) SM is the scrum-master

\(^{37}\) PO is the overall product owner (All names removed to preserve anonymity).
“I just take whatever is first or appropriate to start because there are things that you can’t just start straight away.”

Limk-Developer (QA & Sub-PO)

9.3.4 Post-Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>NPD Goals</th>
<th>New Product Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-10 Post-Sprint - review meeting</td>
<td>NPD-1</td>
<td>E √</td>
</tr>
<tr>
<td>CS-11 Post-Sprint - team retrospective</td>
<td>E √</td>
<td>E √</td>
</tr>
</tbody>
</table>

Table 9-5: NPD goals pursued by practices in ‘Post Sprint’ method fragment (adapted from table 3-11)

Analysis of observations around post-sprint activities identified four NPD goals that motivated these practices: Built-in instability (NPD-1), Self-organization (NPD-2), Multi-level learning (NPD-4) and Organizational Transfer of Learning (NPD-6). Having expressed quite a lot of scepticism in relation to some of the sprint practices, one respondent observed that the requirement to regularly explain incomplete work at the review meeting (CS-10) had motivated him to seek more innovative approaches to addressing his tasks.

“By being pushed harder you figure out how to do stuff in a different way…..I think this was a ticket that went from one sprint to another and I was not finishing and I was being asked why.”

USA-Developer (SE)

The team moved the timing of their retrospective (CS-11) a number of times in order to best meet their own needs. Such autonomy to situate a key practice in the most effective way is an example of self-organization (NPD-2). In relation to the retrospective, they analysed each other’s feedback in order to uncover themes related to potential areas for improvement. As part of the retrospective, they would set action items to alter their work practices and in the following retrospective, they would check to ensure these actions were pursued. This continuous improvement is also evidence of self-organization (NPD-2).

“As a group we bring up a list of improvements…try to come up with 3 themes…improvement was shown with how you could resolve this particular problem… I think this gets people thinking along the right lines”

USA-Developer (QA & Sub-PO)
“Before this we would also look at the themes from the previous retro to see if we acted correctly...”
Limk-Developer (QA & Sub-PO)

The team did not have any external stakeholders at their review meeting. The product owner and team are the only attendees for both the review and retrospective meetings. As a result, although many responses indicated that these practices appeared to pursue a goal of cross-functional learning (NPD-4), they did not appear to seek to share knowledge across the organization. However, the review was useful for both general team awareness of work performed and also to share knowledge on learning achieved in the performance of specific tickets.

“It gives you an overview of what other people are doing but outside of system test, if we didn’t have the review we wouldn’t have a clue what other people are doing.”
Limk-Developer (QA)

“If we had a special challenge with a test case where we were blocked. We do try to address that so that we learn from this.”
USA-Developer (QA)

Both awareness and technical learning are also pursued in the retrospective. Awareness of each other’s views on how best to tackle the work was noted as was the sharing of specific useful work techniques.

“The retrospective is a good eye-opener because it shows the various different perspectives of people when acknowledging that this went wrong or this could be improved.”
USA-Developer (QA)

“If I was investigating something in testcomplete and I found a new fast way to do it that is something that I would put in my ‘something went well’ ”
Limk-Developer (QA & Sub-PO)

9.3.5 Summary of NPD Findings

Many of the characteristics of a successful new product development team are found to be present in the distributed Scrum team. The characteristic that is absent is the organizational
transfer of knowledge (NPD-6). The use of Atlassian JIRA tickets to preserve the history of how completed tickets were performed may serve eventually to achieve this goal. However, in the current evolution of the team, their practices are quite self-contained and this is especially true of the review practice.

Table 9-6 presents an overview of the findings in relation to NPD goals. This summary table is structured in the same manner as the summary tables outlined for goal rationale findings in chapter 8. It is notable that pre-sprint work has led to a large amount of observations related to the pursuit of in-built instability and subtle control. Sprint activities have been most effective in promoting collective responsibility (although it should be noted that this was mostly related to system test duties that form the most widely shared work within the team). Finally, the team currently appears to associate post-sprint work mainly with the pursuit of sharing awareness and learning within the team. However there is a slight overlap here in that some of these findings could be interpreted as the pursuit of cross-fertilization which would reflect a goal of self-organization. It is presumed that as the team evolves and becomes more integrated, the post-sprint activities could potentially become more sophisticated especially in relation to the demonstration of work completed. This could result in a wider audience and greater emphasis placed on this phase, leading to a greater awareness of goals that may be pursued using this method fragment.
<table>
<thead>
<tr>
<th>GENERAL CODE Description</th>
<th>DScrum Goal</th>
<th>Sub-characteristics</th>
<th>Practices</th>
<th>Gen</th>
<th>Pre-sprint Findings</th>
<th>Sprint Findings</th>
<th>Post-sprint Findings</th>
<th>NPD FRAMEWORK EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPD1 - Built-in Instability</td>
<td>g1</td>
<td>Challenging Targets</td>
<td>CS1; CS4; CS7; CS10</td>
<td>2</td>
<td>12*</td>
<td>0</td>
<td>1</td>
<td>Setting challenging targets and releasing control of how these targets should be met introduces a tension and instability that empowers teams, promoting creativity and innovation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Release of control to team</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPD2 - Self Organizing Project Teams</td>
<td>g2</td>
<td>Autonomy</td>
<td>CS1; CS2; CS4; CS7; CS8; CS11</td>
<td>5*</td>
<td>4*</td>
<td>2*</td>
<td>2</td>
<td>Autonomy empowers the team to manage their situation without external interference. Self-transcendence is evident when teams practice continuous improvement via the setting and extending of goals in pursuit of the higher-order challenges set down by senior management. Cross-fertilization occurs as teams share knowledge.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Self-transcendence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cross-fertilization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPD3 - Overlapping Development Phases</td>
<td>g3</td>
<td>Learning across functions</td>
<td>CS1; CS4; CS7; CS8; CS9</td>
<td>1</td>
<td>1</td>
<td>14*</td>
<td>0</td>
<td>promote collective responsibility and the reduction of potential delays. Such delays may be due to bottlenecks and misunderstandings associated with the ‘over the wall’ hand-off approach associated with division of labour using separate specialized development phases</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learning at different levels (individual, group, corp.)</td>
<td>CS1; CS4; CS7; CS10</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>5*</td>
<td></td>
</tr>
<tr>
<td>NPD4 - Multi-level Learning</td>
<td>g4</td>
<td>Freedom within boundaries</td>
<td>CS1; CS2; CS4; CS6; CS8</td>
<td>3</td>
<td>13*</td>
<td>3</td>
<td>0</td>
<td>Subtle control refers to the management of self-organizing teams by permitting freedom within boundaries that are marshalled through non-invasive management techniques that are tolerant and reflective of the uncertain non-linear nature of new product development</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tolerant non-invasive management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPD5 - Subtle Control</td>
<td>g5</td>
<td>Sharing via formal artefacts</td>
<td>CS7; CS10</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Judicious assignment of resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Standardization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9-6: Overview of field results related to presence of NPD goals

* indicates issues with goal achievement;  ! Indicates partial goal motivation;   !! Indicates goal contradiction
9.4 Findings associated with Agile Software Development Principles

The examination of the relationship between ASD principles and Scrum practices uses a similar structure and approach as that presented in the previous section. The hypothesis used to frame the analysis looks for the presence of ASD goals that motivated the execution of Scrum practices in the QUAL-NN project.

Candidate relationships identified between ASD principles and Scrum practices (table 3-11) are contrasted with empirical findings associated with each of the four method fragments under investigation: general considerations, pre-sprint, sprint and post-sprint. As seen in earlier sections, each sub-section commences with a table outlining the candidate goal rationale and the associations uncovered in the empirical research.

### 9.4.1 General Considerations

![Table 9-7: ASD goals pursued by practices in ‘General considerations’ fragment (adapted from table 3-11)]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-1</td>
<td>Scrum Team Configuration</td>
<td>√</td>
<td>√</td>
<td>E</td>
<td>√</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-2</td>
<td>Initial planning (and ongoing) - create or update product backlog</td>
<td>√</td>
<td>E</td>
<td>√</td>
<td>E</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-12</td>
<td>Definition of Done</td>
<td>√</td>
<td>√</td>
<td></td>
<td>E</td>
<td></td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Of the candidate goals, A-2, A-5, A-6 and A-10 were deemed to be present in the empirical findings. Additional goals identified were: A-7 and A-11. Welcoming changing requirements (A-2) is observed as an outcome of the general Scrum structure and iterative nature of the development.

“This was a changing requirement that had not been planned and was never done before. We were happy to do it as another item available for the sprint backlog.”

Limk-Developer (Scrum-Master, QA & Sub-PO)

39 Candidate goal rationale: Each practice-goal association is denoted by a tick mark (√).
Refined goal rationale: Each practice-goal association is denoted by the letter ‘E’ (Empirical finding).
A sense of feeling motivated and trusted (A-5) is evident from one observer’s reflection on the application of the framework. They described how the application of activities that support adherence to a clear definition of done (CS-12) gave them a sense of satisfaction about their work.

“If you can create your tickets in small chunks, chunks small enough to say: this is done when it is done. This is key to finishing work and feeling happy”

USA Developer (SE)

The Fin-Limk team had indicated that the practice of ‘definition of done’ (CS-12) contributed to the pursuit of working software being the primary measure of progress (A-7). This view was echoed by one member of the QUAL team although they acknowledged that due to the fragmented nature of the team’s responsibilities, they had not applied focus to this practice and one might infer that as a result they have not achieved this goal.

“Early on we had a rough discussion of that (definition of done) but as a team I don’t know if it has been elevated to anything of major importance...it is because of limited dependencies”

USA-Developer (QA & Sub-PO)

The definition of done encouraged the pursuit of simplicity (A-10) as it enabled developers to ensure that work was specified in cohesive terms. Clear simple task specification was desired in order to avoid any misunderstandings around the declaration of a ticket to be complete.

“We didn’t want to be carrying over tickets. We wanted tickets refined enough that we could say ‘that’s done’”

USA-Developer (QA & Sub-PO)

Reflections on the general structure and nature of Scrum by developers from each of the remote sites indicate that the framework contributed toward team empowerment and self-organization (A-17).

“The overall structure of Scrum is to be a lot more self-reliant ... to get these tickets completed”

USA-Developer (QA)
“the whole way of organizing the work like this and the fact that we are working together helps us to improve the quality of the work”

USA-Developer (QA)

### 9.4.2 Pre-Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>ASD Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-3</td>
<td></td>
</tr>
<tr>
<td>Sprint planning - create release backlog</td>
<td></td>
</tr>
<tr>
<td>CS-4</td>
<td>E √ E √ E √ E E E E E E</td>
</tr>
<tr>
<td>Sprint planning - create sprint backlog</td>
<td></td>
</tr>
<tr>
<td>CS-5</td>
<td></td>
</tr>
<tr>
<td>Sprint planning - define sprint goal</td>
<td></td>
</tr>
</tbody>
</table>

Table 9-8: Candidate ASD goals pursued by practices in ‘Pre-Sprint’ method fragment (from table 3-11)

With the exception of effective communication (A-6), all other candidate ASD goals were deemed to be evident in the findings associated with pre-sprint practices. Additional goals observed were motivated individuals (A-5) and progress measurement through working software (A-7).

The delivery of customer value (A-1) was promoted by sprint planning practices (CS-4). Tasks were regularly prioritized to meet the needs of internal customers, thus ensuring that value was considered from the customer’s perspective and was delivered early.

“We request the priorities from the teams (which could be viewed as our customers) …they gave us a set of priorities..”

USA-Developer (QA)

Many respondents explained how creation of the sprint backlog (CS-4) for each sprint helped them to welcome changing requirements (A-2). The acceptance of controlled change is clearly described in the following comment:
“From sprint to sprint there is totally different work”
Limk Developer (QA)

Sprint planning (CS-4) served to motivate various members of the team. Self-determination of estimates was highlighted by a number of respondents. They felt trusted and empowered by this practice. It seems that effective planning gave them a sense of achievement.

“It’s good when you do a right estimate. Somehow it feels nice - it feels good to know that you can plan”
USA Developer (SE)

Because much of the work of this group involved supporting compliance and other quality initiatives, it was felt that the definition of the goal of measuring progress by working code (A-7) should be clarified to suit this development situation. This goal is deemed to also measure progress through evidence of completed work that enables code to work. In that sense, the planning and creation of bounded tickets that represent measureable work is considered to be a practice that supports this measurement goal.

“What we wanted to do was create tickets that we would be able to close in this sprint.”
Limk Developer (QA)

A core element of much of sprint planning involved capacity management as developers calculated their available work time in the forthcoming sprint and compared it to the estimates of work commitments being established during the planning meeting. Such practices pursued sustainable development (A-8).

“Just doing time management and being realistic with ticket estimates. You have only 90 hours available for 3 weeks...”
Limk Developer (QA)

“I count my hours, look at availability and make sure it balances”
Limk Developer (SE)

“...try to do some level loading...if there is too much, we will pull it out and put into the following sprint.”
USA Developer (Sub-PO & QA)
Planning the sprint backlog (CS-4) was also seen to pursue simplicity (A-10). Estimation and planning provoked respondents to consider how best to decompose work into manageable portions and consider the simplest approach and solution to an issue. The use of JIRA tickets to support planning served as a structure to help developers preserve their intention to build a simple solution.

“When it is done you should stop and not doing anything more fancy. If it is not in the ticket you should not do any extra work....”

USA Developer (SE)

### 9.4.3 Sprint

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td>E</td>
<td>√</td>
<td>√</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td>√</td>
<td>E √</td>
<td>E</td>
<td>E</td>
<td>E √</td>
<td>E √</td>
<td>√</td>
<td>E √</td>
<td>E √</td>
<td>E √</td>
<td>E √</td>
<td>E √</td>
<td>E √</td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td>√</td>
<td>E</td>
<td>√</td>
<td>E √</td>
<td>E √</td>
<td>E √</td>
<td>√</td>
<td>E √</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td></td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E √</td>
<td>E √</td>
<td>E √</td>
<td>E √</td>
<td>E √</td>
<td>E √</td>
<td>E √</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9-9: Candidate ASD goals pursued by practices in ‘Sprint’ method fragment (from table 3-11)

Sprint findings were consistent with those that emerged from the FIN-NN project (section 9.4.3) in that most of the candidate ASD goals were found to be present in the empirical findings.

The pursuit of simplicity (A-10) was promoted by the regular logging of time by developers. It enabled them to compare the amount of effort expended versus planned on a particular task and use the comparison to avoid over-work on a solution.

“When logging our time against those tasks on a daily basis.. is helpful. I cannot justify that I have spent more than 4 hours on these 2 tasks - what did I do with the other 4 hours? I need to log them somewhere...”

USA Developer (QA)

The daily Scrum (CS-7) was considered to be both motivational (A-5) and a promoter of more effective communication within the team (A-6).
“With the daily Scrum you now know that there is a person on the job and there is interaction ... you hope that inspires that person ...there is a lot more communication so this is a motivational aspect.”

USA Developer (Sub-PO & QA)

“The daily sprint meetings are by design very short...we limit them to 15 minutes so you are not going to have ... people tuning out or not interested. You are focused for that 60 seconds to 2 minutes.”

Limk Developer (Sub-PO & QA)

Another sprint activity that promoted motivation (A-5) within the team was the use of a backlog graph or burndown chart to monitor team progress (CS-9). The regular review of this chart also supported the measurement of progress through completed work (A-7) as it served to remind the team of the importance of accurate time recording.

“Reviewing the daily chart especially when we are getting close to the line - these really motivate you to say “I want to do my part to ensure that we don’t stray from the desired line”......”

USA Developer (Sub-PO & QA)

“At each of the daily meetings, SM\textsuperscript{40} would bring up the burndown chart. Its good - it reminds you to log your time.”

Limk Developer (QA)

A number of the respondents pointed out that occasionally the sprint workload did get modified after sprint work had commenced. It was sometimes done to provide early value to the customer (A-1). This indicates that locking features (CS-6) may contradict the goal of delivering customer value early. However, most comments in relation to unplanned work being imposed upon a sprint noted that the sprint structure combined with the expectation of a locked set of planned work promoted sustainable development(A-8). When unplanned work was proposed, the developers were able to refer to the planned work to either suggest delaying the proposed work or deferring some planned work to a later sprint in order to preserve sustainable development.

\textsuperscript{40} SM: Scrum-Master name omitted to preserve anonymity
“We are able to at least look at what the current workload is, determine if something can be put to the side and then take down the new requirements.”

USA Developer (Sub-PO & QA)

Other practices that promote sustainable development include the daily Scrum (CS-7) and backlog graph (CS-9). The daily Scrum was used to highlight obstacles and collaboration opportunities. Also, on each Friday of the sprint, this meeting was expanded to review the burndown chart and discuss any identified schedule variances.

“... discussion every Friday about the burn-down chart to determine ...why the estimates are not working out ... last week it showed that one member was sick, somebody else took a delayed vacation...”

Limk Developer (QA)

9.4.4 Post-Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>ASD Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-10 Post-Sprint - review meeting</td>
<td>√</td>
</tr>
<tr>
<td>CS-11 Post-Sprint - team retrospective</td>
<td>E</td>
</tr>
</tbody>
</table>

Table 9-10: Candidate ASD goals pursued by practices in ‘Post-Sprint’ method fragment (adapted from table 3-11)

Respondents clearly noted that post-sprint practices were used to pursue developer motivation (A-5) and regular team reflection and improvement (A-12). Other goals observed were effective communication (A-6), progress measurement (A-7), sustainable development (A-8) and simplicity (A-10).

Everybody was in a position to clearly describe the activities that they had to carry out for the retrospective (CS-11). Many different examples were given of how various themes for improvement were suggested and implemented as a result of this practice.

“What went well, didn’t go well and recommendations... I can see what everybody else is writing... it is a shared Google doc... we choose themes. The themes mentioned by everybody are most important.”

Limk Developer (SE)
This team did not have any additional audience members attend the sprint review (CS-10). They also have merged the review and retrospective meetings. However, it is interesting that they distinguished between the retrospective and review practices. As stated above, it is clear that the retrospective promotes team reflection and improvement (A-12). Although there are no external members present, the review appears to motivate the team (A-5) as they do not wish to be seen to fail in their planned work. A very consistent awareness of the importance of this event was recounted by many of the developers, regardless of their core skill set. Statements that express a determination to succeed combined with a fear of failure portray this practice as a very motivational experience. They also reveal a strong collective responsibility within the sub-team that their particular work is perceived as meeting expectations. The positive feedback experienced from success is also noted as delivering satisfaction to the team.

“Creating tickets .. knowing that you have to get done is a great motivator.. review meeting and if you don’t have all your tickets closed - you know that other people are going to be looking at your work..”
Limk Developer (QA)

The retrospective practice (CS-11) was supported by pre-work using a shared Google document to promote effective communication (A-6) between team members. This document served as the agenda for this meeting and ensured that there was tight focus and effective decision making in their efforts to synthesize improvement suggestions into three common themes. An improvement applied to this practice was the decision to have each team member read out their suggestions rather than having an up-front section to the meeting where attendees read the entire document. This modification to the meeting improved momentum and collaboration.

“Everybody would have the Google doc pre-filled. ...we talk individually about our own because it is a better flow to the meeting rather than having a silent period of 10 minutes while people are reading”
Limk Developer (SE)

Reflections on the review meeting also note that it has promoted the measurement of progress through fully completed tasks (A-7). There had been a tendency for tasks to be partially completed and remain open as a result of dependency issues such as development environments.
The review meeting (CS-10) has increased the visibility of these problems as developers are not allowed to close a ticket unless all of the work is complete and they must explain the reasons for any incomplete tickets at this meeting. This has led to outcomes such as the re-specification of tickets so that the completed portion was closed and a separate ticket opened for the remaining work. One might infer that this will lead to more effective decomposition of tasks in future sprints.

“There may be trouble test cases that due to either an environment or some test execution issue...you close out what has been done and leave open only that element that is still open and carry that forward”

Limk Developer (SE)

9.4.5 Summary of ASD Findings

Table 9-11 presents a high-level overview of responses provided in relation to ASD goals that are motivated by Scrum practices. This table is merely an indication of the responses that provoked most reflection. One possible use of this information is to note the goals that provoked the greatest number of responses within the various method fragments. The detailed reflections described in this section (9.4) are of much greater use as they provide a rich description of the views of individual team members on different Scrum practices.

The asterisk and exclamation mark indicators are also used in this table to denote goal achievement, partial motivation and contradiction issues.

<table>
<thead>
<tr>
<th>GENERAL CODE Description</th>
<th>DScrum Goal</th>
<th>Practices</th>
<th>Gen</th>
<th>Pre-sprint Findings</th>
<th>Sprint Findings</th>
<th>Post-sprint Findings</th>
<th>ASD FRAMEWORK EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASD1 - Satisfy customer</td>
<td>g7</td>
<td>CS4; CS6;</td>
<td>0</td>
<td>2</td>
<td>1*</td>
<td>0</td>
<td>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</td>
</tr>
<tr>
<td>ASD2 - Changing requirements</td>
<td>g8</td>
<td>CS2; CS4;</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.</td>
</tr>
<tr>
<td>ASD3 - Frequent software delivery</td>
<td>g9</td>
<td>CS7;</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
</tr>
<tr>
<td>ASD4 - Business &amp; developer collaborate</td>
<td>g10</td>
<td>CS7; CS8</td>
<td>0</td>
<td>0</td>
<td>3*</td>
<td>0</td>
<td>Business people and developers must work together daily throughout the project.</td>
</tr>
<tr>
<td>ASD5 - Motivated individuals</td>
<td>g11</td>
<td>CS1; CS4; CS7; CS9; CS10; CS11; CS12</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>10</td>
<td>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</td>
</tr>
<tr>
<td>ASD6 - Face-to-face communication</td>
<td>g12</td>
<td>CS2; CS7; CS9; CS11</td>
<td>1*!!</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</td>
</tr>
<tr>
<td>ASD7 - Working software is measurement of progress</td>
<td>g13</td>
<td>CS4; CS7; CS8; CS9; CS10; CS12</td>
<td>1*</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>Working software is the primary measure of progress.</td>
</tr>
<tr>
<td>ASD8 - Sustainable Development</td>
<td>g14</td>
<td>CS4; CS6; CS7; CS9; CS10; CS11</td>
<td>0</td>
<td>20*</td>
<td>10</td>
<td>2</td>
<td>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</td>
</tr>
<tr>
<td>ASD9 - Technical excellence and good design</td>
<td>g15</td>
<td>CS8;</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Continuous attention to technical excellence and good design enhances agility.</td>
</tr>
<tr>
<td>ASD10 - Simplicity</td>
<td>g16</td>
<td>CS4; CS7; CS10; CS12</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>Simplicity – the art of maximizing the amount of work not done – is essential.</td>
</tr>
<tr>
<td>ASD11 - Self-organizing teams</td>
<td>g17</td>
<td>CS1; CS7;</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>The best architectures, requirements, and designs emerge from self-organizing teams.</td>
</tr>
<tr>
<td>ASD12 - team reflection and improvement</td>
<td>g18</td>
<td>CS7; CS11</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6*</td>
<td>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.</td>
</tr>
</tbody>
</table>

Table 9-11: Overview of field results related to presence of ASD goals

---

* Indicates issues with goal achievement; ! Indicates partial goal motivation; !! Indicates goal contradiction
9.5 Findings associated with GSD Challenge Alleviation Goals

The third hypothesis proposed to structure exploration of this situation declared:

“A distributed team that uses Scrum will overcome many GSD challenges”.

The structure applied to the presentation of findings in chapter 9 is also used in this section. Findings have been interpreted for evidence of the three GSD challenge alleviation goals and these are presented in four sections representing practices from each of the method fragments.

In order to enrich the descriptions of Scrum practices being used to alleviate GSD findings, this section follows the approach adopted in section 8.5. Each GSD challenge alleviation finding is denoted by a suffix to indicate the form of distance (temporal, geographical or socio-cultural) reduced by the Scrum practice applied.

In each of the following sections, empirical findings are contrasted with the proposed findings that informed construction of the candidate LDScrum model. As seen in the NPD and ASD sections, each sub-section begins with a table contrasting the candidate and empirical goal rationales for the practices of the fragment.

9.5.1 General considerations

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Scrum Practices</th>
<th>GSD1 Communication</th>
<th>GSD2 Coordination</th>
<th>GSD3 Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-1</td>
<td>Scrum Team Configuration</td>
<td>✓/ ✓/ ✓</td>
<td>E ✓</td>
<td>✓</td>
</tr>
<tr>
<td>CS-2</td>
<td>Initial planning (and ongoing) - create or update product backlog</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CS-12</td>
<td>Definition of Done</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 9-12: Candidate GSD Challenge Alleviation goals pursued by practices in ‘General Considerations’ method fragment (adapted from table 4-10)

Candidate goal rationale: Each practice-goal association is denoted by a tick mark (✓).
Refined goal rationale: Each practice-goal association is denoted by the letter ‘E’ (Empirical finding).
Coordination challenge alleviation goals emerged as the main goals pursued by general practices. Temporal distance was alleviated by tailoring of the pre-sprint and post-sprint meeting structure. Initially the team held the review and retrospective on the last day of the sprint and the planning meeting was scheduled for the following Monday (first day of the next sprint). Due to the eight hour temporal distance between the teams, both meetings were compressed into one longer meeting. The reason for this structure was to alleviate the GSD coordination challenge of having reduced hours for synchronous collaboration (GSD2-T).

"Retrospective and review on Friday and planning on a Monday…those guys were coming in at 6…rather than getting them in at that time 2 consecutive days, we lengthened the meeting and do it all in one day.”
Limk Developer (Sub-PO & QA)

A number of references were made to the increased sense of unity and teamness that has been promoted by use of the Scrum framework. The geographical distance between team members could be a threat to unity but various comments from team members from both locations indicated that the Scrum practices (CS-1) have improved matters in this regard (GSD2-G).

"Reviews and the planning meeting .. gives good visibility into the amount of work that the team actually does which is something you would not have if you had 7 or 8 people there working on their own.”
Limk Developer (Sub-PO & QA)

"It reminds you that you are a team on a regular basis…all of it builds in a team aspect right from the planning meeting through the daily meetings to the retrospective. All of this is a team participation activity.”
USA Developer (Sub-PO & QA)
9.5.2 Pre-Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>GSD Challenge</th>
<th>GSD1 Communication</th>
<th>GSD2 Coordination</th>
<th>GSD3 Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-3</td>
<td>Sprint planning - create release backlog</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CS-4</td>
<td>Sprint planning - create sprint backlog</td>
<td>✓</td>
<td>E</td>
<td>✓</td>
</tr>
<tr>
<td>CS-5</td>
<td>Sprint planning - define sprint goal</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 9-13: Candidate GSD Challenge Alleviation goals pursued by practices in ‘Pre-Sprint’ method fragment (adapted from table 4 10)

The pre-sprint fragment was highlighted by one respondent as a useful mechanism to overcome the difficulty in initiating contact with remote colleagues within the team. Delays caused by this communication challenge were alleviated as sprint planning (CS-4) enabled the team member to become aware of the different competencies and duties of colleagues (GSD1-G).

“Simply because we have a clearly defined team ... we have a clear delineation of who is doing what activity. In JIRA we can find out who is doing a particular task.”

USA Developer (Sub-PO & QA)

The sprint planning meeting (CS-4) enabled remote team members to discuss interdependencies that might exist between tasks and help coordinate the work to be done during the sprint. This helped to overcome the challenge of coordination complexity associated with distributed teams (GSD2-G).

“Each person discusses the issues and she opens it up for questions after each discussion. There has been dialogue. Sometimes there is interdependencies”

USA Developer (Sub-PO & QA)

One respondent explained that the sprint planning meeting (CS-4) was in place to get everybody together in order to promote a sense of unity among the distributed team (GSD2-G). However their comments indicated that this goal is not yet being fully achieved due to the division of responsibilities that are embedded within the team.
“It’s basically getting all the players in one room talking about what they are going to be doing. There are people that are focussing on the tasks that they have and it really has no bearing on me.”

USA Developer (Sub-PO & QA)

9.5.3 Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>GSD Challenge</th>
<th>GSD1 Communication</th>
<th>GSD2 Coordination</th>
<th>GSD3 Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td>GSD1-T Temporal</td>
<td>GSD1-Geographical</td>
<td>GSD1-Socio-Cultural</td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td>GSD2-T Temporal</td>
<td>GSD2-Geographical</td>
<td>GSD2-Socio-Cultural</td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td>GSD3-T Temporal</td>
<td>GSD3-Geographical</td>
<td>GSD3-Socio-Cultural</td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9-14: Candidate GSD Challenge Alleviation goals pursued by practices in ‘Sprint’ method fragment (adapted from table 4.10)

The majority of recollections of GSD challenges being alleviated relate to sprint practices. The daily Scrum (CS-7) was considered effective in helping to reduce the problems associated with many GSD challenges. Communication issues due to temporal, geographical and socio-cultural distance are alleviated. Delays due to a small overlap of working hours between locations (GSD2-T) are reduced by having an established and mandatory communications link each day.

“I think the daily meeting gives us that communication point where if we have anything to bring and it does ensure that we try to have overlap.”

USA Developer (QA)

Because all team members are expected to be present at the daily Scrum and to report on their general activities, the increased effort to initiate contact (GSD1-G) was reduced:

“You know what is expected. You know that you have to be there with your update”

USA Developer (QA)

Communication between geographically distant team members can also be hampered by reliance on technical infrastructure (GSD1-G). The structure of the daily Scrum enables clear facilitation of the meeting by leveraging technology effectively. One participant in the daily Scrum
explained how cumbersome and frustrating it could be if each member has to pass presenter rights to the next contributor in a meeting that uses application sharing technology. However, he felt that this communication issue had been removed by the effective facilitation and use of the technology by the Scrum-master. What this excerpt reveals is that by having a Scrum-master and a meeting structure that mandates the approach taken by each contributor, use of the technology may be enhanced.

“SM has her finger on the trigger there and when I finish talking she has given it to the next person. It is pretty snappy and people’s updates are snappy as well”

Limk Developer (Sub-PO & QA)

Different socio-cultural backgrounds can lead to communication misunderstandings due to the lack of a common frame of reference (GSD1-C). These were alleviated as the structure of the daily Scrum (CS-7) clearly dictated the information to be communicated.

“It takes out ..room for interpretation ... what you did yesterday, impediments, and what you plan to do tomorrow .. Regardless of where you are from, that is what you have to come in with and you get your 1 or 2 minutes to provide that.”

Limk Developer (Sub-PO & QA)

One team member explained that the regular frequent communication and coordination afforded by the daily Scrum helped the distributed team overcome the challenge of coordination complexity (GSD2-T). It also helped her to retain a sense of focus when parts of the communication related to areas outside her domain knowledge.

“If we weren’t doing Scrum, we were only talking to the US folks once a week in one hour meetings and your mind might wander off.... In the 15 minutes meeting , you know it’s for your own group and that is more effective...”

USA Developer (Scrum-master, Sub-PO & QA)

When most of the team were involved in the system test work, they indicated that the standing system test meeting (CS-7) called immediately following the daily Scrum was very effective and helped to overcome coordination complexity (GSD2-T). Many respondents identified this

43 SM: scrum master name is removed to preserve anonymity.
meeting as being most relevant and effective to their work. However, the general daily Scrum meeting is also highlighted as a practice that promoted a sense of awareness and team spirit (GSD2-G) among the distributed team.

“There is huge visibility between team members and it improves the relationships’ between members because meeting all the team helps us become friends.”

Limk Developer (SE)

“Daily sprint meetings allow us to all be together at the same time and discuss issues. Gives us a better visibility as to what they are doing in Limerick.”

USA Developer (QA)

A challenge facing distributed teams that are geographically distributed is the reduced chances for informal interaction (GSD1-G). One respondent described the daily Scrum as an opportunity for remote colleagues to be part of a conversation which may not directly impact them at the time but could help them deal with a related situation in the future. Whereas this view appeared to be accepted when discussing the system test work shared by most of the team, it was not dominant in relation to other team activities. One respondent in particular did not see the potential for informal interactions through the daily Scrum due to the division of responsibilities within the team. However, one would hope that this benefit would become more evident as the team’s workload becomes more integrated.

“It is very helpful (for a distributed team) as we might have an issue and they might not have seen it yet but when they hit it they will be able to recognize it. We are helping one another out and it saves time”

Limk Developer (Scrum-master, Sub-PO & QA)

### 9.5.4 Post-Sprint

<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>GSD Challenge</th>
<th>GSD1 Communication</th>
<th>GSD2 Coordination</th>
<th>GSD3 Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>GSD1-T Temporal</td>
<td>GSD1-G Geographical</td>
<td>GSD1-C Socio-Cultural</td>
</tr>
<tr>
<td>CS-10</td>
<td>Post-Sprint - review meeting</td>
<td>E</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CS-11</td>
<td>Post-Sprint - team retrospective</td>
<td>√</td>
<td>√</td>
<td>E</td>
</tr>
</tbody>
</table>

Table 9-15: Candidate GSD Challenge Alleviation goals pursued by practices in ‘Post-Sprint’ method fragment (adapted from table 4 10)
Communication within distributed teams is reliant on effective tools and environments (GSD2-T). Using JIRA issue management software running on a corporate network, the team was well-equipped to effectively communicate. This is most evident in the post-sprint practice of review (CS-10). Using the AT&T application sharing software, each team member was able to present their work within JIRA. The application presented the work as ‘cards’ so that all other team members are able to see their colleague’s tasks presented in the same structure as post-it notes on a traditional information radiator such as a Scrum board.

“Typically we do the review first. At that point each screen share and talk about our own respective tickets using JIRA. There is a cards feature when you look at the screen you see all the tickets you have with three columns, work to-do and done. Scrum-master can control presenter rights very quickly.”
Limk Developer (Sub-PO & QA)

“We all share our screen - go through the JIRA tickets in the greenhopper classic task board. So we go through the list of closed, what is still open and whether anything needs to be deferred”
USA Developer (QA)

The retrospective (CS-11) also promoted teamness and unity (GSD2-G). One reflection described how communication processes using the daily Scrum were addressed in the retrospective leading to a more effective and inclusive approach.

“Another thing that came out of it (RETROSPECTIVE) would have been on our daily Scrum meetings.. we also go into greater detail on our tickets. We also do the screen share...”
Limk Developer (Sub-PO & QA)

9.5.5 Summary of GSD Findings

Table 9-16 presents an overview of the findings in relation to practices performed in order to alleviate GSD challenges. This table reveals that respondents more readily identified with instances where coordination challenges were addressed (GSD-2). Of 55 responses, just 4 related to communication (GSD-1) and 7 were associated with control (GSD-3).
Of the four method fragments explored, the majority of responses (27) relate to sprint practices and 15 to post-sprint, leaving just 13 observations from the other two fragments.

Responses were gleaned from inquiries into different sub-themes of each of the three major GSD challenges. This additional granularity was used in order to increase the accuracy of goal association to different practices. Therefore, if a response indicated that a sprint practice was performed to overcome language difficulties, this would be coded as a socio-cultural communication challenge. This table shows the finding as GSD1-C to reflect a reduction in socio-cultural distance. For the purposes of establishing the Scrum goal rationale, findings from all three distances are synthesized to the higher-order goal. Results at this level may be seen in table 9-16 by reviewing the total row for GSD1, GSD2 and GSD3. This table is also used to verify data presentation as the totals are reconciled with raw data findings to confirm that all findings have been covered in the presentation.
<table>
<thead>
<tr>
<th>CODE</th>
<th>DScrum Code</th>
<th>Practices</th>
<th>Gen</th>
<th>Pre-sprint</th>
<th>Sprint</th>
<th>Post-sprint</th>
<th>Specific challenge</th>
<th>GSD FRAMEWORK EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD1-T</td>
<td>G19</td>
<td>CS7</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>Delayed Communication</td>
<td>It may take longer to contact somebody when they are needed. There may be delayed response due to misunderstandings and it may take longer to resolve entire issue due to delayed feedback.</td>
</tr>
<tr>
<td>GSD1-G</td>
<td>G19</td>
<td>CS7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Lack of informal communication</td>
<td>Reduced chance of informal communication - reduced opportunity to build working relationships. Reduced flow of info. on changes and loss of essential aspect of design activities. High dependency on effective tools and environments. May be variances in available technology. Increased costs due to necessity for travel. Difficult to determine remote skillsets and who to contact</td>
</tr>
<tr>
<td>GSD1-C</td>
<td>G19</td>
<td>CS7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Managing frames of reference</td>
<td>Language misunderstandings. Also need to overcome different frames of reference in communication.</td>
</tr>
<tr>
<td>GSD2-T</td>
<td>G20</td>
<td>CS1;CS7</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td>Reduced Hours of collaboration</td>
<td>Misunderstandings &amp; rework can cost. Reduced hours of collaboration. Difficult to have synchronous team meetings. Downtime of needed technologies Coordination can be costly due to amount of non-routine work - needs close management</td>
</tr>
<tr>
<td>GSD2-G</td>
<td>G20</td>
<td>CS1;CS4;CS7;CS11;CS7</td>
<td>3</td>
<td>1*</td>
<td>7</td>
<td>1</td>
<td></td>
<td>Reduced Trust</td>
</tr>
<tr>
<td>GSD2-C</td>
<td>G20</td>
<td>CS8</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>Doubtful of other’s abilities</td>
<td>May have doubts of their colleagues abilities May have conflicting views of domain knowledge May require language and cultural training</td>
</tr>
<tr>
<td>GS02</td>
<td>G20</td>
<td></td>
<td>5</td>
<td>3</td>
<td>16</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSD3-T</td>
<td>G21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Management of project artefacts</td>
<td>Issue with completion of processes between sites. Requirements mods, poor tools, lack of informal contact impact concurrent engineering</td>
</tr>
<tr>
<td>GSD3-G</td>
<td>G21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Lack of concurrent engineering principles</td>
<td></td>
</tr>
<tr>
<td>GSD3-C</td>
<td>G21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Perceived threat from low-cost alternatives</td>
<td>Perceived threat of loss of job to lower-cost colleague Need to be aware of local norms and practices Different perceptions of authority</td>
</tr>
</tbody>
</table>

Table 9-16: Overview of field results related to presence of GSD challenge alleviation goals

\(^*\) indicates issues with goal achievement; \(!\) Indicates partial goal motivation; \(!!!\) Indicates goal contradiction
## 9.6 Findings associated with Lean Software Development Values

<table>
<thead>
<tr>
<th>LDScrum Goal</th>
<th>Code</th>
<th>Description</th>
<th>LDScrum Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>g1</td>
<td>NPD-1</td>
<td>Built-in instability</td>
<td></td>
</tr>
<tr>
<td>g2</td>
<td>NPD-2</td>
<td>Self-organizing project teams</td>
<td></td>
</tr>
<tr>
<td>g3</td>
<td>NPD-3</td>
<td>Effectively Overlap development phases</td>
<td></td>
</tr>
<tr>
<td>g4</td>
<td>NPD-4</td>
<td>Multi-level learning</td>
<td></td>
</tr>
<tr>
<td>g5</td>
<td>NPD-5</td>
<td>Leverage Subtle control over work tasks</td>
<td></td>
</tr>
<tr>
<td>g6</td>
<td>NPD-6</td>
<td>Disseminate learning through organization</td>
<td></td>
</tr>
<tr>
<td>g7</td>
<td>A-1</td>
<td>Satisfy the customer: early val. software.</td>
<td></td>
</tr>
<tr>
<td>g8</td>
<td>A-3</td>
<td>Welcome changing requirements.</td>
<td></td>
</tr>
<tr>
<td>g9</td>
<td>A-2</td>
<td>Deliver working software frequently</td>
<td></td>
</tr>
<tr>
<td>g10</td>
<td>A-4</td>
<td>Business and developers work closely.</td>
<td></td>
</tr>
<tr>
<td>g11</td>
<td>A-5</td>
<td>Build projects around motivated individuals.</td>
<td></td>
</tr>
<tr>
<td>g12</td>
<td>A-6</td>
<td>Face-to-face communication</td>
<td></td>
</tr>
<tr>
<td>g13</td>
<td>A-7</td>
<td>Working software as measure of progress.</td>
<td></td>
</tr>
<tr>
<td>g14</td>
<td>A-8</td>
<td>Sustainable development</td>
<td></td>
</tr>
<tr>
<td>g15</td>
<td>A-9</td>
<td>Continuous attention to technical excellence</td>
<td></td>
</tr>
<tr>
<td>g16</td>
<td>A-10</td>
<td>Simplicity.</td>
<td></td>
</tr>
<tr>
<td>g17</td>
<td>A-13</td>
<td>Self-organizing teams.</td>
<td></td>
</tr>
<tr>
<td>g18</td>
<td>A-12</td>
<td>Regular Team reflection</td>
<td></td>
</tr>
<tr>
<td>g19</td>
<td>GSD-1</td>
<td>Overcome GSD communication challenges</td>
<td></td>
</tr>
<tr>
<td>g20</td>
<td>GSD-2</td>
<td>Overcome GSD coordination challenges</td>
<td></td>
</tr>
<tr>
<td>g21</td>
<td>GSD-3</td>
<td>Overcome GSD control challenges</td>
<td></td>
</tr>
</tbody>
</table>

Table 9-17: Summary of candidate LDScrum value rationale

This section presents the value rationale found to be present within the QUAL-NN project. Each LSD value is examined and the reasons why the pursuit of certain goals is deemed to be underpinned by the value are explained. The same approach and structure used in section 8.6 to explain the value rationale within the FIN-NN project is applied here. Furthermore, there is an element of duplication in this section as the presentation of a number of tables is repeated in order to enhance readability.
In order to assist in the comparison of the candidate value rationale to the empirical findings, this section opens with a table outlining the candidate value rationale and concludes with a table summarizing the refined value rationale. The opening table 9-17) below is a duplicate of table 8-17 but is repeated here as it is considered useful to encapsulate this content within the same section.

Sub-sections 9.6.1 to 9.6.12 associate each of the twelve LSD values with the goals that they are seen to underpin based upon the empirical evidence. In order to support the reasoning used to make these associations, each sub-section opens with a table describing the candidate value-goal associations proposed in chapter 5. Each of these twelve tables has already been presented in their respective LSD value sub-sections of section 8.6. However, they are duplicated here as they contain the description of the principles that comprise the LSD value and are intended to be a practical reference for the reader when reviewing value rationale explanations.

Please note that the previous sections used the prefix ‘NPD-n’, ‘A-n’ and ‘GSD-n’ to denote NPD, ASD and GSD goals respectively. This section uses the mapping of these goals to their LDScrum goal code (g1 - g21).

### 9.6.1 ‘V1’ Customer Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer value</td>
<td>Ensure customer defines value. Seek out user needs, not just requirements. Early delivery of value. Enable future needs (maintainability). Enable efficient deployment of product features. Provide value for money.</td>
<td>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</td>
</tr>
</tbody>
</table>

Table 9-18: Candidate Value Rationale for V1 - ‘Customer Value’

The candidate table proposed that customer satisfaction (g7) is the sole goal underpinned by customer value (V1). Analysis of empirical data indicated that g7 was pursued as a result of this value, but it also uncovered a number of other goals that seemed to have been influenced by V1:
• Built-in Instability (g1)
• Subtle Control (g5)
• Completed work (working software) is the primary measure of progress (g13)
• Sustainable Development (g14)
• Simplicity (g16)

In describing an aspect of built-in instability (g1) associated with their work as having challenging targets, one developer explained how a prioritization process was needed to deal with the various requirements. This explanation revealed an underlying value to ensure early delivery of value to the customer.

“There is loads of requirements coming in...I am supposed to be doing 90 hours and my first estimate was 140 hours..take this bit off, do this bit differently so that we could cut back the hours...”

Limk Developer (SE)

Pursuit of the goal to measure progress through completed work (g13) was influenced by customer value also as one developer explained how the review aims to determine if certain activities need to be rolled forward to the next sprint.

“We go through each issue that we have had in the previous sprint and determine are we actually done - is it complete or am I rolling it forward..”

USA Developer (Sub-PO & QA)

A few developers described that although sustainable development was pursued (g14), occasionally business and customer pressures dictate that there was a need to take on unsustainable levels of work. This reveals an underlying value to provide customer value (V1). The pursuit of satisfying the customer was influenced by this value. In one example, a developer explained how the goal of satisfying the customer was driven by ensuring that the various internal customers (software development teams) participate in the prioritization of tasks for the sprint. Another reflection explains how the goal of satisfying the customer leads to the team breaking the practice of locking features in order to provide early delivery of value.
“We request the priorities from the teams (which could be viewed as our customers).”
USA Developer (QA)

“Things can get thrown away in the middle of the sprint by PO."
Limk Developer (Scrum-master, Sub-PO & QA)

The pursuit of simplicity (G16) was described in the context of customer’s requirements and needs by one developer. This example shows how the goal was influenced by ‘Customer Value’ and also to a lesser extent by ‘Business Environment Awareness’ (V8). Please note that in the following quote, the respondent’s first language was not English and their use of the word ‘should’ was intended to be ‘could’.

“Whenever we are doing the planning for the sprints there would be so many things that we should (sic.) do but we will only pick whatever we are supposed to do and that is really important”
Limk Developer (Scrum-master, Sub-PO & QA)

### 9.6.2 ‘V2’ Reduce Waste

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce Waste</td>
<td>Promote systematic defect prevention. Eliminate rework through emphasis on customer needs. Holistic focus on quality that motivates the refactoring of legacy code to improve general product. Identify silos of waste present in value stream and improve processes. Promote reusability where appropriate. Take cognisance of the fact that waste can have different forms in different projects/work situations.</td>
<td>Simplicity - the art of maximizing the amount of work not done - is essential</td>
</tr>
</tbody>
</table>

Table 9-19: Candidate Value Rationale for V2 - ‘Reduce Waste’

The candidate value rationale proposes that simplicity is the only goal underpinned by this value. However, although waste reduction (V2) was found to influence simplicity, it was also found to be present in the pursuit of two other goals:

- Motivated and trusted developers given suitable environments (g11)

45 PO: This is the overall product owner - name removed to preserve anonymity
- Sustainable Development (g14)

Two respondents explained that they tended to confine their tasks to work that they had estimated and committed to doing at the outset of the sprint. The underlying value here was to reduce waste. In the case of one respondent, they explained that their pursuit of simplicity (g16) was to avoid having to describe overwork or wasteful activities at the sprint review.

"Logging our time against those tasks on a daily basis...if you say I cannot justify that I have spent more than 4 hours on these 2 tasks..."

USA Developer (QA)

Another respondent provided an example of how the daily Scrum motivated him to be more productive in his work performance (g11). The underlying value present here was to reduce waste caused by a lack of focus and attention due to general procrastination.

"Daily sprints probably help because you are forced to speak of your progress or lack of - if you only give one update a week, you’re disguised by the fact that you don’t have to give an update for another week."

Limk Developer (Sub-PO & QA)

An observation on the pursuit of sustainable development (g14) through the management of developer capacity and estimated workloads appeared to be primarily underpinned by effective process (V10) as it indicated an appreciation for a predictable scalable approach to work. However, waste reduction (V2) was also evident here in relation to the concern for developer burn-out and the consequent waste due to productivity reduction.

"We are tackling availability and we try to measure the estimated time against our availability so that we try to make sure that we don’t burn people down by over assigning work to them."

Limk Developer (Sub-PO & QA)
9.6.3 ‘V3’ Flow of Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g9</th>
<th>Goal - g14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow of Value</td>
<td>Stabilize development process to enable levelled flow of value - manage necessary variability. Smooth demand from users. Pull value from user demands through lifecycle. Apply minimalism (scope items, teams, documents). Allocate resources only when needed.</td>
<td>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
<td>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely</td>
</tr>
</tbody>
</table>

Table 9-20: Candidate Value Rationale for V3 - ‘Flow of Value’

Of the two candidate goals proposed, sustainable development (G14) was the only one identified in the findings. However, a number of additional goals were found to be underpinned by V3:

- Motivated and trusted developers who are given the environment they need (g11)
- Completed work (working software) is the primary measure of progress (g13)
- Simplicity (g16)

The practice of using the daily Scrum to support planned activities motivated developers to manage their work effectively. Such motivation was achieved as a result of regular progress control measures that ensured productivity did not deteriorate. This helped to avoid a hectic push to complete many tasks near the end of the iteration. This approach to smooth out demand reveals the influence of ‘Flow of Value’.

“If you are not doing it right you could end up working like a mad person in the last week.....they do (discuss progress in daily Scrum) and there is no mad rush really.”

Limk Developer (SE)

Whereas the point above focussed on tasks within a sprint, flow of value (V3) may also be considered from the broader perspective of how value flows from sprint to sprint. One developer explained how using the sprint review to examine completed work achieved a goal of measuring progress through completed tickets (g13). This reflection reveals that ‘Flow of Value’ was an
influence on the pursuit of this goal as the goal was used to deliver completed work and also plan additional work required to deliver related value in the ensuing sprint.

“The common consensus is that there is almost always something that is left over... you close out what has been done and leave open only that element that is still open and carry that forward.”

Limk Developer (SE)

The decomposition of work into clear tickets has helped the team to pursue sustainable development (g14). The desire to reduce variability in order to complete activities reveals the influence of ‘Flow of Value’ on the pursuit of this goal. This observation was made by a number of developers although in some cases, the dominant value was interpreted to be effective process (V10). This will be discussed further in section 9.6.10.

“I think we are getting more done. Before Scrum, the work was not broken down in tickets. You break things down a bit more to ensure that things are getting done.”

Limk Developer (Scrum-master, Sub-PO & QA)

The majority of responses relating to the influence of this value referred to the goal of simplicity (g16). Pursuing simplicity through the decomposition of work into clear tasks in the planning stage was influenced by V3. This activity shows an appreciation of the management of variability through the application of minimalism to scope.

“When you are doing an estimate for a task you are working on the basis of making it as short as possible - having it smaller rather than larger because you only have a small amount of time”

Limk Developer (Scrum-master, Sub-PO & QA)

One observation indicated that early delivery of value (V1), flow of value (V3) and effective process (V10) were all seen to underpin the pursuit of simplicity. The developer explained that when planning tickets, they occasionally found that the amount of work required may exceed the capacity available. However, they were not in a position to completely postpone internal customer requests to a future sprint (when they would have available capacity). They needed to decompose the work further into simpler chunks so that they could fulfil a piece of each customer’s needs in order to avoid customer work being suspended due to dependencies on the
QUAL-NN project. Ultimately V3 was deemed to be the dominant influence due to the respondent’s desire to keep a flow of value going through to different customers, even when it was not possible to satisfy all their needs.

“You can’t really ever have a situation where you say I can’t do anything ..with this thing for the next 3 weeks because I am full. You would have to do some chunk on it to keep that ball in the air. The challenge then is to say how much can I do there to keep me within my available hours.”

Link Developer (Sub-PO & QA)

9.6.4 ‘V4’ Person Focus

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g2</th>
<th>Goal - g5</th>
<th>Goal - g11</th>
<th>Goal - g17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person Focus</td>
<td>Empower individuals and teams. Promote continual worker development. Invest in multi-skilling and promote attitude to participate in multiple roles.</td>
<td>Self-organizing project teams. (Autonomy, Self-transcendence, Cross-fertilization)</td>
<td>Leverage Subtle control over work tasks</td>
<td>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</td>
<td>The best architectures, requirements, and designs emerge from self-organizing teams</td>
</tr>
</tbody>
</table>

Table 9-21: Candidate Value Rationale for V4 - ‘Person Focus’

Interpretation of QUAL-NN findings revealed instances where all four proposed goals were found to be influenced by ‘Person Focus’. A further five goals also appeared to be associated with this value:

- Built-in Instability (g1)
- Overlapping Phases - collective responsibility (g3)
- Multi-level Learning (g4)
- Sustainable Development (g14)
- Alleviate GSD coordination challenges (g20)

The pursuit of in-built instability (g1) was afforded by the sprint structure especially in the sprint planning meeting where new requirements could emerge at each iteration. Many respondents described how they were empowered to review estimates and discuss their tasks in this meeting.
In the current approach, estimates and tasks have been assigned to developers prior to the meeting. However the pre-assignment activities often involve informal meetings between the Sub-PO and the developers in that sub-team and it was in these meetings where it was most evident that developers were empowered, thus revealing this underlying value.

“It could happen in the sub-team meeting that morning where QA1 and QA2 may disagree that a ticket should be put into the next sprint.”

Link Developer (Sub-PO & QA)

The review practice threw up a lot of issues around non-closure of certain items. Pursuit of in-built instability (g1) within the environment was evident from challenging workloads and the demand for closure of these tasks provoked innovation. Person focus (V3) underpins this goal as the review process was being used to promote continual worker development by determining reasons for perceived productivity problems.

“By being pushed harder you figure out how to do stuff in a different way.....I think this was a ticket that went from one sprint to another and I was not finishing and I was being asked why.”

USA Developer (SE)

Most of the team expressed a clear view that the sprint review promoted a sense of motivation (g11) within each person as they recounted their achievements in the sprint. This goal exhibits a value of person focus (V3) as developers felt responsible for their tasks as a result of being empowered to complete specific work items.

“You don't want to look bad and appear to look like a slacker so you don't want to have to explain why you are behind. There is a level of motivation there.”

USA Developer (QA)

The association of this value to the pursuit of motivated developers (g11) was also evident in the following recollection of the daily Scrum practice.

46 QA1 & QA2: These are two developers in the same sub-team. Both names are removed to preserve anonymity.
“The ownership ... we are encouraged to have this thing throughout the Scrum meetings we have by showing that I have done this - this is completed now”

Limk Developer (SE)

The pursuit of self-organization (g2) is underpinned by person focus as the team lengthened their sprint duration in order facilitate team development. They felt that retrospectives were occurring too rapidly and the modified structure appeared to improve their work.

“I think we are more intense and more focused now …. We started off as a 2 week sprint but ... the retrospectives were happening way too quickly so we moved to a 3 week sprint.”

USA Developer (Sub-PO & QA)

Developers were encouraged to help one another in tasks so that there could be more flexibility in planning and management of work. The following excerpt explains how some software engineering work was being shared with another developer whose core skill set was as a QA. The sharing of this work aimed to achieve the goal of overlapping development phases (g3) through the cross-training of developers.

“She can do parts of it. The parts she is getting to do is so that she can help me out with some of the work.”

Limk Developer (SE)

At the time of writing, a more explicit form of developer training was also underway in order to increase the breadth of skills within the team. While this activity is intended to enable the pursuit of overlapping development phases (g3) during development, it was a very clear training activity and as such could be considered the pursuit of multi-level learning (g4). Encouragement of developers to participate in multiple roles indicates the value of person focus in this situation.

“All the quality group members have just gone through that training to be internal auditors. I see definite improvements here. QA1 and QA2 are both in the training cycle as well.”

USA Developer (Sub-PO & QA)

47 QA1 & QA2: two developers in different sub-teams. Their names are removed to preserve anonymity.
Developers were empowered to set their targets within the confines of both the sprint duration and their capacity to meet their workload. The pursuit of subtle control (g5) was clearly underpinned by person focus as both developers and the team were empowered to determine how best to complete the work.

“These are the tickets, this is the time you have been assigned to meet those tickets - it is our own choice on how we are going to resolve tickets.”

Limk Developer (SE)

A number of responses explained that sprint planning was an issue for the team. Estimation of tasks was an ongoing challenge that required improvement. The pursuit of sustainable development (g14) was reliant on better estimation and many of the responses reveal an underlying person focus value as they show a clear preference for individual developers to drive their own estimation needs. This reflects both developer empowerment and growth.

“Time management and being realistic with ticket estimates. You have 90 hours available for 3 weeks...It’s not until you realistically write things down that you can see that you are not going to get it done!”

Limk Developer (QA)

The general manner in which the Scrum framework was being applied by compressing the review and retrospective of the completed sprint into the same meeting as the planning meeting for the forthcoming sprint helped to alleviate the GSD coordination challenge of having reduced hours for synchronous collaboration (g20). The value that was predominant in the pursuit of this goal was effective collaboration (V11), but there was also an element of team empowerment present which reveals a value of person focus (V4).

“It was a Monday morning but it did not work for anybody. The US folks are happy with an early start on Friday as they like to finish early on that day”

Limk Developer (Scrum-master, Sub-PO & QA)
9.6.5 ‘V5’ Continuous Improvement

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g1</th>
<th>Goal - g4</th>
<th>Goal - g6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Improvement</td>
<td>Visible feedback on productivity. Reviews to propagate learning to wider organizations. Apply rigorous standardization to establish agreed baselines for improvement. Actively seek and manage obstacles through agreed processes, such as root cause analysis. Promote organizational learning and a culture of relentless improvement. Use simple powerful tools.</td>
<td>Built-in instability</td>
<td>Multi-level learning</td>
<td>Effectively disseminate learning throughout Organization</td>
</tr>
</tbody>
</table>

Table 9-22: Candidate Value Rationale for V5 - ‘Continuous Improvement’

Two of the candidate goals were found to be underpinned by this value: g4 and g6. Four additional goals were also found to be influenced by continuous improvement (V5):

- Self-organization (g2)
- Completed work (working software) is the primary measure of progress (g13)
- Sustainable Development (g14)
- Regular reflection and improvement (g18)

The influence of V5 on the goal of linking completed tickets to progress measurement (g13) is evident through the focus of visible feedback on productivity issues. The team review (CS-10) of open tickets at sprint completion enabled root cause analysis to be undertaken and improvements to be put in place.

“The JIRA tickets show a great visibility of specific tasks and the work that you did on them and the way you did them”

USA Developer (QA)

“We might have 10 scripts to fix up. Once they are all fixed up, we close the ticket. If we don’t get all 10 done, we may have to list it as an impediment because we may be waiting for a technical resource...”

Link Developer (Scrum-master, Sub-PO & QA)

A related goal was the pursuit of sustainable development (g14). Having to explain on a regular basis (each sprint review), how tickets were completed and why certain items were not completed should help to build better planning skills and achieve sustainable development.
“We would all go through the tickets from the sprint just finished and say go through the ‘done’ tickets. We would then go through the tickets we got half-done and explain why they were not completed...”
link Developer (Scrum-master, Sub-PO & QA)

There are a number of recollections on how the retrospective (CS-11) achieved the goal of regular reflection and improvement (g18). The influence of continual improvement (V5) on the pursuit of this goal is seen through various examples that portray a growing culture of relentless improvement.

“In one retrospective, when we started out with Scrum one of the main themes was that we were still a bit confused about logging time in JIRA. Things like that would come out and be resolved”
link Developer (QA)

“There is two types of feedback - ... sprint practices we are using and ... the workload we are doing...we collate those ... 3 different themes and ...assign somebody to take action on those for the next sprint..”
link Developer (Sub-PO & QA)

Another goal pursued by the retrospective practice (CS-11) was self-organizing teams (g2). As part of sprint planning, the team has incorporated corrective action items from the retrospective performed in the previous sprint. This root cause analysis of obstacles and the removal of same through corrective action management reflect a continual improvement (V5). The following excerpt explains how the team used the sprint retrospective to ensure that corrective action plans had been executed as planned.

“To take it back one step, before this we would also look at the themes from the previous retrospective to see if we acted correctly..”
link Developer (Sub-PO & QA)

Multi-level learning (g4) was also pursued by the retrospective (CS-11) and review (CS-10). Retrospective information was held in an internal document. The team also used the review to understand discrepancies between original estimates and actual time expended on a ticket.

“... to address if there is a huge discrepancy between the original estimate and the time logged. If we had a special challenge with a test case where we were blocked... so that we learn from this...”
USA Developer (QA)
JIRA tickets contain comment lines that developers may have filled in to describe innovative solutions applied to an issue. Any such comments were read out during reviews. The distinction between the aforementioned retrospective internal document and these JIRA tickets was accessibility. Because the tickets may be viewed by other departments within I-E-S (such as product support), they were a training resource. Having such information available to propagate knowledge throughout the organization (g6) reflects a value of continual improvement (V5).

“There are comment areas on the JIRA tickets and we would go through the comments if we wanted to at the review. So I might say I had hit this blocker and here is the reason I got around it…”

USA Developer (QA)

9.6.6 ‘V6’ Product Excellence

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Excellence</td>
<td>Promote strong design culture to avoid premature convergence on incorrect solution. Prototype different options to derive best approach. Deliver functionality in change-tolerant form. Build quality in through practices of jidoka and poka-yoke. Promote deep specialized knowledge of product and process. Promote a culture of excellence.</td>
<td>Continuous attention to technical excellence and good design enhances agility.</td>
</tr>
</tbody>
</table>

Table 9-23: Candidate Value Rationale for V6 - ‘Product Excellence’

Very little emerged from the findings in relation to this value. This may be due to the diverse nature of the work performed by the group or because they perceive themselves as a services group to other teams within product development. The only recollection deemed to be influenced by this value was the pursuit of in-built instability (g1). In this example, the developer explained how the team argued and discussed various work items in order to avoid premature convergence upon a solution.

“We have had multiple team debates about different things ... we have had discussions about certain tickets and certain things we don’t always agree on”

USA Developer (Sub-PO & QA)
9.6.7 ‘V7’ Embrace Change

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g1</th>
<th>Goal - g8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embrace Change</td>
<td>Defer commitment to scope. Encapsulate features and consider all options carefully. Facilitate emerging requirements.</td>
<td>Built-in instability</td>
<td>Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.</td>
</tr>
</tbody>
</table>

Table 9-24: Candidate Value Rationale for V7 - ‘Embrace Change’

As one would expect from its title, this value is found to underpin welcoming changing requirements (g8). No evidence was found to support the other candidate association (g1). However, an additional goal emerged from the findings:

- Sustainable Development (g14)

The creation of a clear task plan for each sprint supported effective management of change requests. An underlying value to embrace change underpins the pursuit of welcoming changing requirements. In the following reflection, the developer explained that although they planned to lock features for a sprint, it was still possible that they might be asked to apply immediate focus to a customer need that could emerge in the middle of a sprint. In such cases, they were able to assess how to prioritize and re-plan in order to cater for this intrusion. A secondary value evident in this example was effective process (V10) as the sprint plan and daily Scrum enabled assessment of the team’s productivity rate in order to determine the impact of an emerging requirement.

“Because you have thought about all the tasks and laid out the time for them...You still have to take on those changes ... but you are armed with more information...how you can structure and organize it.”

Limk Developer (Sub-PO & QA)

In her consideration of the broader development roadmap, the Scrum-master described how the emergence of new requirements was facilitated by a continual grooming of the product backlog. Such acceptance of change reveals an underlying value of V7.
“PO and I would meet every week... about what needs to be done over 6 months... break them down in priority. If anything extra comes up, PO would let me know or I would let PO know.”

Link Developer (Scrum-master, Sub-PO & QA)

Another example of this value was seen in this recollection of the addition of new requirements to a sprint backlog (CS-4) although they had not been part of the long-term roadmap.

“There was this new ‘on-demand’ stuff... a changing requirement that had not been planned and was never done before. We were happy to do it as another item available for the sprint backlog”

Link Developer (Scrum-master, Sub-PO & QA)

The practice of locking features (CS-6) for the sprint helps to enable predictable planning and sustainable development. However, as stated earlier, new work can emerge during the sprint. This finding explains how team members were dealing with new issues by raising additional tickets. In order to sustain development (g8), it was necessary that the impact of the inclusion of additional work be considered by the team. The underlying value here was to embrace change (V7) but also effective collaboration (V11) through the promotion of teamwork.

“A few people were deciding to raise tickets without discussing it with anybody...we said that we would bring it back into line so that we would not do any extra work without discussing it with the team first.”

Link Developer (Scrum-master, Sub-PO & QA)

9.6.8 ‘V8’ Business Environment Awareness

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Environment Awareness</td>
<td>Be aware of why you are doing work. Ensure it adds value to your situation. Future-proof your approach to work against changing market conditions. Consider domain solutions rather than restricting product to one market sector. Don’t force lean approach where inappropriate to business needs. Don’t optimize locally to the detriment of other aspects of the business.</td>
<td>Business people and developers must work together daily throughout the project.</td>
</tr>
</tbody>
</table>

Table 9-25: Candidate Value Rationale for V8 - ‘Business Environment Awareness’

48 PO: overall product owner name is removed to preserve anonymity
Pursuit of the sole candidate goal (G10) was found in the recollection of one developer about maintaining a good working relationship with the I-E-S support group. This was influenced by an underlying value to have ‘Business Environment Awareness’ so that they know the reasons behind certain work requirements.

“They are adding bits and pieces. By me keeping in touch with them all the time and keeping the gate open I think that they are happy and that we have a good relationship between quality and support.”

Limk Developer (Scrum-master, Sub-PO & QA)

Two other goals emerged that appeared to be influenced by this value:

- Welcome changing requirements (g8)
- Simplicity (g16)

The overall product owner (and manager of general quality activities within I-E-S) attended the daily Scrum meeting on a very regular basis. This practice was motivated by the goal to have developers meet business people daily (g9) which was influenced by a need to have awareness for why work was being done and confirmation that it added value to the situation (V8). Furthermore, it is interesting that this comment was made by a developer based in the USA as they would not have had much opportunity to informally interact with the Irish based quality manager.

“Our primary customer is PO⁴⁹ and he participates in the daily meetings on a fairly regular basis.”

USA Developer (Sub-PO & QA)

A particular reflection that had associated the practice of task selection within sprint-planning (CS-4) to be the pursuit of simplicity was deemed to be underpinned by ‘Customer Value’ (V1). However this description is also influenced by business environment awareness (V8) as the team sought to only do work that added value to the customer.

⁴⁹ PO: overall product owner name is removed to preserve anonymity.
“Whenever we are doing the planning for the sprints there would be so many things that we should do but we will only pick whatever we are supposed to do and that is really important.”

Limk Developer (SE)

### 9.6.9 ‘V9’ Data-driven Decisions

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Driven Decisions</td>
<td>Impartial data collection to drive decision making and reduce cost of meetings and disagreements. Rigorous scientific approach to continuous improvement.</td>
<td>No goals Identified</td>
</tr>
</tbody>
</table>

Table 9-26: Candidate Value Rationale for V9 - ‘Data-driven Decisions’

No goal-value associations emerged from the findings.

### 9.6.10 ‘V10’ Effective Process

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g13</th>
<th>Goal - g18</th>
<th>Goal - g21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Process</td>
<td>Easily understood method. Enable clear comprehension of method-in-action. Clear mechanisms for resource allocation including consistent roles, names, work practices. Ability to assess team productivity rate. Central expert to integrate overall development (‘Chief Engineer’). Scalable approach. Clear product development roadmaps. Fast powerful feedback loops to build momentum.</td>
<td>Working software is the primary measure of progress.</td>
<td>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.</td>
<td>Overcome GSD control challenges</td>
</tr>
</tbody>
</table>

Table 9-27: Candidate Value Rationale for V10 - ‘Effective Process’

This value was seen to influence the pursuit of many of the Scrum goals. In some instances, a more dominant value may have been evident but effective process (V10) still appeared to be a factor in the goal being sought. All three candidate goals were identified as were many others:

- Built-in Instability (g1)
- Self-organization (g2)
- Subtle Control (g5)
- Welcome changing requirements (g8)
- Regular collaboration between developers and business people (g10)
- Sustainable Development (g14)
- Continuous attention to technical excellence (g15)

One of the reasons why so many reflections exhibit elements of this value is because the team was in the early part of their adoption of Scrum and they had placed a very large emphasis on understanding their development capacity. They addressed the challenging targets associated with built-in instability (g1) by understanding their ability to deal with differing requirements (CS-4). This particular excerpt indicates the achievement of this goal being underpinned by a value to have an effective process where the team productivity rate was assessed and used to ensure effective work performance.

“There was a lot more focus on the tasks ... a larger awareness in the noise that is coming in from outside. Somebody will come in with a special request from outside and "ah that really has a big impact on my time!" ... makes it a lot more intense.”

Limk Developer (SE)

Estimation of planned work was another aspect of the sprint planning process (CS-4) used to address in-built instability (g1). Person focus (V4) was deemed to be the dominant value that underpinned this goal. Developers were empowered to review tasks and estimates that had been assigned to them in JIRA tickets. However, because tasks and estimates were pre-assigned by the sub-PO of each area, person-focus did not appear to be as influential as was observed with the Fin-Limk team. The Fin-Limk sprint planning meetings leveraged the story points technique which appeared to be a more inclusive approach to work planning. The reason that the QUAL team used an approach of ticket allocation was because the team was still at an early stage in their adoption of Scrum. Earlier attempts to have developers enter their own estimates in JIRA failed due to poor appreciation for the importance of this activity to planning and capacity management. As a result, ticket estimates were pre-assigned and reviewed thus providing some empowerment and consideration for person-focus (V4). This innovative planning approach emerged from the instability of challenging targets within a flexible environment (g1). The fact
that there was central integration of resource allocation and work management indicates that at that point in the evolution of their Scrum adoption, effective process (V10) was influential in their pursuit of this goal.

“She would already have assigned us tickets that day so we would have kind of known what tickets had been assigned to us. So, Sub-PO\textsuperscript{50} would put time estimates on those...”

Limk Developer (QA)

This central integration of work (V4) also arises in relation to the pursuit of sustainable development (g14). An aspect of the pre-print planning (CS-4) activities involved the comparison of available developer capacity to the estimates associated with the sprint tasks. Central resource allocation is evident in the planning stage to address overcapacity situations. Furthermore, the Scrum-master had introduced an extra meeting within the sprint to determine whether the team were still on track in relation to the original estimates.

“90 hours of work (15* 6) ... something wrong if you have planned 160 hours ... PO\textsuperscript{51} will be keeping a running total 90 v 70; 90 v 120 ... where we are in terms of overallocation and under allocation...”

Limk Developer (Sub-PO & QA)

“...about a week or a week and a half in, SM\textsuperscript{52} would go around the table and ask where are we in terms of your current workload and what you might have to deliver by the end of the sprint...”

Limk Developer (Sub-PO & QA)

The pursuit of simplicity (g16) in the prioritization of work was also observed to be influenced by ‘Effective Process’ due to expert integration of the work. This value was also seen in the pursuit of regular reflection and adjustment (g18). One developer explained that the daily Scrum (CS-7) enabled them to make adjustments to priorities or work approaches as needed (most likely as a result of productivity assessments).

\textsuperscript{50} Sub-PO: Sub-product owner name omitted to preserve anonymity
\textsuperscript{51} PO: product owner name is removed to preserve anonymity
\textsuperscript{52} SM: scrum-master name is removed to preserve anonymity
“The daily meetings.. there is a good amount of knowledge shared there and we make adjustments throughout the sprint. We either prioritize the tickets or work through them more effectively.”

USA Developer (QA)

The pursuit of self-organization (g2) through cross-fertilization occurred as the team shared knowledge due to their use of the Scrum framework. The underlying value here appears to relate to effective process through enhanced communication and integration of activities. This appears to be especially true when respondents described system test work as this was the work that engaged team members from the three different sub-teams.

“We have a larger visibility of quality across product. I feel the communication during system test is very good and valuable.”

USA Developer (QA)

System test was cited by many developers in relation to the pursuit of collective responsibility (g3). The daily Scrum (CS-7) enabled the system test sub-team to overlap their work and collaborate in achieving the overall goal. Use of the burndown chart (CS-9) prompted reallocation of work in order to bring things back on line. This is an example of overlapping development phases where certain team members might have been finished their assigned script executions due to particular specialist knowledge but they still made themselves available to ‘muck in’ and assist with work that had proved problematic for colleagues. The value of effective process (V10) is evident in the pursuit of this goal as the team had an ability to assess their productivity rate and used a central expert to integrate the work.

“..We would bring up the burn-down chart and see where we are. Within the system test..she might see that somebody has all their work done and somebody has not and she would automatically divvy it out.”

Limk Developer (Sub-PO & QA)

The pursuit of subtle control (g5) involves having freedom within boundaries. The team pursued this goal in the manner in which work was planned and allocated. In one example, the Sub-PO took control of sprint backlog item selection (CS-4). Her knowledge of the area combined with feedback from the product owner gave her the freedom to determine how best to hit targets. As stated earlier, sprint-planning (CS-4) would exhibit greater freedom within boundaries if the
team were more heavily involved. The problem was that when the team were initially given more freedom, they failed to enter estimates. In order to address this issue, it was decided to reduce the freedom by having the Sub-PO drive sprint-planning. This reveals a value of effective process to enable resource allocation and integration of developer activities.

“...Before the sprint begins, I would look at what needs to be done e.g. there might be 5 things out of the list of 20 that I think can be done in this sprint...”

Limk Developer (Sub-PO & QA)

Another variation of the pursuit of subtle control (g5) is seen in how the team addresses requirements that emerge mid-sprint. The core practice of locking features (CS-6) was accepted as an aspiration but a subtle control was put in place to manage the reality of mandatory interruptions. A bucket of time was planned and formalized within a JIRA ticket for work that was unknown at the outset of the sprint. The value here was effective process as there was a clear comprehension of the method-in-action.

“There is unforeseen problems that we have to work on sometimes on a daily basis. One thing that is good now is that PO\(^{53}\) created a ticket for whatever is not planned - we can enter hours against that ticket...”

USA Developer (SE)

9.6.11 ‘V11’ Effective Collaboration

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal - g3</th>
<th>Goal - g12</th>
<th>Goal - g19</th>
<th>Goal - g20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Collaboration</td>
<td>Reduce distance between collaborators and processes. Simple visual measures of productivity and quality. Balance functional expertise with cross-functional integration. Promote teamwork. Integrate suppliers. Unambiguous communication procedures.</td>
<td>Effectively Overlap development phases</td>
<td>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</td>
<td>Overcome GSD communic. challenges</td>
<td>Overcome GSD coord. challenges</td>
</tr>
</tbody>
</table>

Table 9-28: Candidate Value Rationale for V11 - ‘Effective Collaboration’

\(^{53}\) PO: Product owner name omitted to preserve anonymity
All four goals proposed to be underpinned by this value in the candidate LDScrum model were present in the findings. More findings were associated with this value than any other. However it should be noted that almost half of all findings were associated with the two candidate GSD challenge alleviation goals - g19 and g20. A number of additional goals were found to be underpinned by effective collaboration:

- Built-in Instability (g1)
- Self-organization (g2)
- Regular collaboration between developers and business people (g10)
- Motivated and trusted developers given suitable environments (g11)
- Sustainable Development (g14)
- Best work emerges from self-organizing teams (g17)

A consistent theme of teamwork promotion underlies many of the goals pursued. The pursuit of in-built instability (G1) is intended to provoke a tension that promotes innovation and teamwork. In her reaction to the challenging targets that face the team, one developer observed that they were almost unrealistic. In order to succeed, she often relied on assistance from colleagues.

“We are not being realistic I think. If I have a problem I can’t resolve, I can check with QA54 or email the group. So it’s a collaborative thing because some people may have more experience on something...”
Limk Developer (QA)

When developers collaborate to address tasks that may be outside of their core specialty, self-organization through cross-fertilization (g2) may occur as they expose their skills and experiences to one another. The underlying value of collaboration that promotes the pursuit of this goal can be seen to be even more effective (V11) when considered from the perspective of remote team members.

54 QA: developer name is removed to preserve anonymity
“Let’s say somebody from the US is struggling to fix an issue that I have worked on this morning ... then I can help. The 15 minutes meetings improve the communication and brings the team together ...”

Limk Developer (SE)

As one might expect, many of the findings reveal that the candidate goal of overlapping development phases (g3) is influenced by this value. These observations cover three main collaborative themes: knowledge sharing; teamwork and collective responsibility to the completion of work. Most of the reflections in this regard refer to the systems test activities as these form the most visible opportunity for a large group within the overall team to share in the achievement of a common goal.

“... in the communal tasks such as system test or the work that QA, SE and I are on, there is a sharing of stuff...”

Limk Developer (Sub-PO & QA)

The core practices of sprint planning (CS-4) and retrospective (CS-11) are performed using rigorous and unambiguous procedures. These meetings leverage shared spreadsheets to support effective communication (g12). The following excerpt explains how the group modified the disciplined communication procedures of the retrospective in order to improve collaboration. They had originally permitted the team to spend 10 minutes of the meeting privately reading each other’s contribution to the shared spreadsheet. This approach was altered as they felt that collaboration was improved by having each person read out their submissions.

“The document should be filled out...we used to take a few minutes for people to read ...now we talk individually about our own because it is a better flow to the meeting ...we try and derive 3 themes...”

Limk Developer (Sub-PO & QA)

The unambiguous procedures associated with the daily Scrum also support the objective of having clear communication. JIRA tickets are used to support work descriptions and employment of strict time restrictions and well-understood core questions ensure the meeting

\[55\] QA & SE are two developers on the same sub-team. Their names are removed to preserve anonymity.
was crisp. In the context of distributed teams, these procedures serve as a common reference for all team members and help to overcome GSD challenges (g19).

“Daily sprint meetings are by design by intention very short...limit them to 15 minutes so you are not going to have somebody giving a 10 minutes update and people tuning out or not interested...”

Limk Developer (Sub-PO & QA)

Other GSD communication issues may arise due to delays caused by difficulties determining who to contact in relation to different issues. The value of effective collaboration (V11) is seen to underpin the desire to alleviate such problems (g19). The team plan their work using JIRA tickets and support task execution with daily communication to reinforce awareness of work responsibilities within the distributed team.

“We have a clearly defined team and we do communicate on a daily basis in a formal meeting. ..we have a clear delineation of who is doing what activity. In JIRA we can find out who is doing a particular task.”

Limk Developer (Sub-PO & QA)

A key principle associated with effective collaboration (V11) is the reduction of distance between collaborators. Many of the respondents commented on the reduction of temporal distance enabled by core meetings being scheduled in the US morning and Irish afternoon in order to leverage the brief overlap of working hours (g20).

“The daily meeting gives us that communication point.. I don’t overlap by much unless the Irish folks tend to work late sometimes so the daily meeting forces that communication and it gives us one opportunity.”

USA Developer (QA)

Compression of the review (CS-10) and retrospective (CS-11) into the same meeting as the planning meeting (CS-4) for the forthcoming sprint helped to alleviate the GSD coordination challenge of having reduced hours for synchronous collaboration. This reduction in temporal distance indicates an underlying value of effective collaboration.
“We had been doing the retro and the review on Friday and the planning on a Monday...we are 8 hours ahead of the other folks and those guys were coming in at 6 for these planning meetings so rather than getting them in at that time 2 consecutive days, we lengthened the meeting and do it all in one day.”

Limk Developer (Sub-PO & QA)

The Scrum-master explained that the daily Scrum reduced geographical distance by affording the team a regular opportunity to discuss work. Her comments noted that this practice might potentially mimic the accidental coordination that occurs from informal interactions within a collocated team.

“It is very helpful (for a distributed team) as we might have an issue and they might not have seen it yet but when they hit it they will be able to recognize it. We are helping one another out and it saves time.”

Limk Developer (Scrum-master, Sub-PO & QA)

Another coordination challenge associated with geographical distance is a loss of unity or sense of ‘teamness’. A number of reflections commented on how Scrum practices helped to reduce this distance.

“Having a distributed team is new to me. I do like getting a chance to hear all of the voices at least once a day. It keeps us connected.”

USA Developer (QA)

The pursuit of sustainable development (g14) motivated the team to introduce regular reviews of the burndown chart (CS-9). This chart was a focal point for discussions on the need to improve either on planning or performance. Simple visual measures of productivity reveal an underlying value of effective collaboration.

“Looking at it every day gives you an indication if you are going off track. Before we were only doing it once a week...and we were going “why is the chart looking like that!”. Now we can see earlier …”

Limk Developer (QA)

Becoming a self-organizing team (g17) is reflected upon by one developer who observed that performing work using the general Scrum framework promoted teamwork (which indicates that effective collaboration underpins the pursuit of this goal).
“The whole way of organizing the work like this and the fact that we are working together helps us to improve the quality of the work.”

Limk Developer (SE)

9.6.12 ‘V12’ Effective Use of Technology

<table>
<thead>
<tr>
<th>Value</th>
<th>Value Description</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective use of technology</td>
<td>Technology must serve a particular need. Configuration management systems and quality measurement devices are examples of key facilities. Pursue autonomation by using technology to enable workers to effectively adjust system when necessary.</td>
<td>No goals Identified</td>
</tr>
</tbody>
</table>

Table 9-29: Candidate Value Rationale for V12 - ‘Effective Use of Technology’

Although no candidate goals were associated with the effective use of technology, the findings indicated the presence of this value (V12) in the pursuit of four goals. The value was considered to be the main influence on a number of recollections describing practices where GSD challenges were alleviated.

- Alleviate GSD communication challenges (g19)
- Alleviate GSD coordination challenges (g20)

Both GSD communication and coordination challenge alleviation goals were achieved by leveraging network technologies and collaborative applications to facilitate meetings between the distributed sites. The observation below describes how these technologies enable JIRA to be used to overcome communication challenges (G19) as desktop sharing enabled all distributed members to view tasks in the same manner as a collocated group working with a traditional Scrum board.

“AT&T s/ware allows you to share out your desktop and as I am talking about my tickets the folks in California or anybody else in the room can see my 3 columns: to-do, in-work and done and I will be talking about them...”

Limk Developer (Sub-PO & QA)
Earlier findings described how a regular daily Scrum meeting helped to alleviate GSD coordination challenges (g20). The following reflection describes how the Scrum-master used technology to control this meeting (CS-7) across a distributed team by swiftly switching presenter rights between participants as required.

“Dial into AT&T.. I go first and go through my list of tickets..then I give presenting rights to the next person and they show their list of tickets and go through any they are working on.”

Limk Developer (Scrum-master, Sub-PO & QA)

In some other recollections, effective use of technology (V12) was evident in aspects of the findings but it was not deemed to be the main influence on the goal being pursued. The goals identified were:

- Self-organization (g2)
- Organizational transfer of learning (g6)

The team exhibited autonomy (g2) in the manner that they conducted sprint planning (CS-4). Their preparation and assignment of tickets using Atlassian JIRA enabled developers to review their work and empowered them to determine whether the estimates could be met. The main value underpinning this work is person-focus (V4) but the effective use of technology (V12) is also present here as JIRA is leveraged to facilitate this process.

“We had the issues that we were planning on assigning...already in JIRA so that everybody could see them and start do the level loading...ensure that the appropriate amount of work was being assigned...”

USA Developer (Sub-PO & QA)

The use of JIRA to document solutions to various sprint requirements pursued organizational transfer of knowledge (g6). These formal documents were accessible throughout I-E-S and can be used by staff to understand aspects of quality related to the various projects that comprise the internal customer base of QUAL-NN. The primary value associated with this goal is continuous improvement (V5). The use of an accessible and well-understood issue-tracking system to facilitate knowledge propagation indicates the influence of effective use of technology (V12).
“I put comments in the JIRA tickets...They are in JIRA so everybody can see in to the JIRA tickets (including the wider organization).”

USA Developer (SE)

9.6.13 Summary of LSD Findings

Table 9-30 presents an overview of how responses were interpreted in order to uncover LSD value associations with goals pursued by the team. The structure of this table is identical to the LSD findings summary in chapter 8 (table 8-30). Parentheses indicate secondary values. An asterisk indicates instances where the value is deemed to have had a partial influence on pursuit of the goal. Exclamation marks highlight situations where the pursuit of a goal contradicted a particular value.

Effective collaboration and person-focus are seen to be the LSD values most prevalent among the findings. This may be as a result of a highly fragmented team aiming to leverage the Scrum framework to improve resource-sharing. This certainly appears to be the case with regards to their system test work. Effective process is another value that appears to underpin many of the findings. This may also relate to the fact that the team is using Scrum to improve their work approach and become more effective at predicting their capacity and outputs.
<table>
<thead>
<tr>
<th>LDScrum Code</th>
<th>General Code Description</th>
<th>All Goals</th>
<th>Gen</th>
<th>Gen Goals</th>
<th>Pre-sprint</th>
<th>Pre-Sprint Goals</th>
<th>Sprint</th>
<th>Sprint Goals</th>
<th>Post-sprint</th>
<th>Post-sprint Goals</th>
<th>General Code Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Customer value</td>
<td>g1,g5,</td>
<td>0</td>
<td></td>
<td>6</td>
<td>g1,g5,g7,g14,g16</td>
<td>1</td>
<td>g7</td>
<td>1</td>
<td>g13</td>
<td>Ensure customer defines value. Seek out user needs, not just requirements. Early delivery of value. Enable future needs (maintainability). Enable efficient deployment of product features. Provide value for money.</td>
</tr>
<tr>
<td>V2</td>
<td>Remove waste</td>
<td>g11,</td>
<td>1 (1)</td>
<td>g16,</td>
<td>2</td>
<td>g11, g16,</td>
<td>0</td>
<td>g11, g16,</td>
<td>0</td>
<td>g16</td>
<td>Promote systematic defect prevention. Eliminate rework through emphasis on customer needs. Holistic focus on quality that motivates the refactoring of legacy code to improve general product. Identify silos of waste present in value stream and improve processes. Promote reusability where appropriate. Take cognisance of the fact that waste can have different forms in different projects/work situations.</td>
</tr>
<tr>
<td>V3</td>
<td>Flow of value</td>
<td>g11,</td>
<td>1</td>
<td></td>
<td>7 (5)</td>
<td>g13, g14, g14, g16</td>
<td>1</td>
<td>g11</td>
<td>2</td>
<td>g13, g16</td>
<td>Stabilize development process to enable levelled flow of value - manage necessary variability. Smooth demand from users. Pull value from user demands through lifecycle. Apply minimalism (scope items, teams, documents). Allocate resources only when needed.</td>
</tr>
<tr>
<td>V4</td>
<td>Person focus</td>
<td>g1,g2,</td>
<td>25</td>
<td></td>
<td>10</td>
<td>g1, g2, g5, g11, g14</td>
<td>12</td>
<td>g1, g4, g6, g11</td>
<td>0</td>
<td>g1, g4, g6, g11, g14</td>
<td>Empower individuals and teams. Promote continual worker development. Invest in multi-skilling and promote attitude to participate in multiple roles.</td>
</tr>
<tr>
<td>V5</td>
<td>Continuous Improvement</td>
<td>g4,g13</td>
<td>4</td>
<td>g4, g13</td>
<td>6</td>
<td>g6, g13</td>
<td>15*</td>
<td>g2, g4, g6, g13, g14, g18</td>
<td>Visible feedback on productivity. Reviews to propagate learning to wider organizations. Apply rigorous standardization to establish agreed baselines for improvement. Actively seek and manage obstacles through agreed processes, such as root cause analysis. Promote organizational learning and a culture of relentless improvement. Use simple powerful tools.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V6</td>
<td>Product excellence</td>
<td>g1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>g1</td>
<td></td>
<td>g1</td>
<td></td>
<td></td>
<td>Promote strong design culture to avoid premature convergence on incorrect solution. Prototype different options to derive best approach. Deliver functionality in change-tolerant form. Build quality in through practices of jidoka and poka-yoke. Promote deep specialized knowledge of product and process. Promote a culture of excellence.</td>
</tr>
<tr>
<td>V7</td>
<td>Embrace change</td>
<td>g8, g14</td>
<td>2</td>
<td></td>
<td>6</td>
<td>g8</td>
<td>5</td>
<td>g14</td>
<td></td>
<td></td>
<td>Defer commitment to scope. Encapsulate features and consider all options carefully. Facilitate emerging requirements.</td>
</tr>
<tr>
<td>LDScrum Code</td>
<td>General Code Description</td>
<td>All Goals</td>
<td>Gen</td>
<td>Gen Goals</td>
<td>Pre-sprint</td>
<td>Pre-Sprint Goals</td>
<td>Sprint</td>
<td>Sprint Goals</td>
<td>Post-sprint</td>
<td>Post-sprint Goals</td>
<td>General Code Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------</td>
<td>-----------</td>
<td>-----</td>
<td>-----------</td>
<td>------------</td>
<td>-----------------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>V8</td>
<td>Business environment awareness</td>
<td>(g16),g9, g10</td>
<td>2</td>
<td>(g16)</td>
<td>2</td>
<td>g9, g10</td>
<td>Be aware of why you are doing work. Ensure it adds value to your situation. Future-proof your approach to work against changing market conditions. Consider domain solutions rather than restricting product to one market sector. Don’t force lean approach where inappropriate to business needs. Don’t optimize locally to the detriment of other aspects of the business.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V9</td>
<td>Data driven decisions</td>
<td>(g1),g2, g5, (g5), (g8), (g14), (g15)</td>
<td>4</td>
<td>(3)</td>
<td>20*</td>
<td>(g1),g3,g5, g7,g8, g14, g14, g16</td>
<td>7</td>
<td>(1)</td>
<td>(g3),g5, (g10), g14, g5, g18</td>
<td>(g18)</td>
<td>Easily understood method. Enable clear comprehension of method-in-action. Clear mechanisms for resource allocation including consistent roles, names, work practices. Ability to assess team productivity rate. Central expert to integrate overall development (‘Chief Engineer’). Scalable approach. Clear product development roadmaps. Fast powerful feedback loops to build momentum.</td>
</tr>
<tr>
<td>V10</td>
<td>Effective process</td>
<td>(g1),g2, g3, g10, g11, g12, g14, (g14), g17, g19, g20, (g20)</td>
<td>7*</td>
<td>(1)</td>
<td>9</td>
<td>g12, g17, g20</td>
<td>41*(2)</td>
<td>(g14), (g20), g2, g3, g10, g14, g12, g14, g19, g20</td>
<td>g11,g12, g20</td>
<td>Reduce distance between collaborators and processes. Simple visual measures of productivity and quality. Balance functional expertise with cross-functional integration. Promote teamwork. Integrate suppliers. Unambiguous communication procedures.</td>
<td></td>
</tr>
<tr>
<td>V11</td>
<td>Effective collaboration</td>
<td>(g1), g2, g3, (g10), g11, g12, g14, (g14), g17, g19, g20, (g20)</td>
<td>7*</td>
<td>(1)</td>
<td>9</td>
<td>(g14), g1, g2, g14, g19, g20, g20*</td>
<td>3</td>
<td>g11, g12, g20</td>
<td>g19, g20, g20* (g6)</td>
<td>g19</td>
<td>Technology must serve a particular need. Configuration management systems and quality measurement devices are examples of key facilities. Pursue automation by using technology to enable workers to effectively adjust system when necessary.</td>
</tr>
<tr>
<td>V12</td>
<td>Effective use of technology</td>
<td>(g2), (g6), g19, g20</td>
<td>1</td>
<td>(1)</td>
<td>2</td>
<td>g19</td>
<td>Technology must serve a particular need. Configuration management systems and quality measurement devices are examples of key facilities. Pursue automation by using technology to enable workers to effectively adjust system when necessary.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>25</td>
<td>79</td>
<td>79</td>
<td>35</td>
<td>218</td>
<td>218</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note - * means that value not fully underpinning goal. !! Means that value is contradicted by this goal. (xx) means that this goal is secondarily underpinned by value and has been counted already in a primary value. Do not count (xx) in totals as they are secondary associations.

Table 9-30: Overview of all LSD findings
9.7 Refined LDScrum Model (Scrum of Scrums)

This chapter presented the findings from analysis of the use of the Scrum framework within the QUAL-NN project. The project was performed by the QUAL team who are a distributed team configured in a ‘Totally Integrated Scrum’ structure. In order to support future related research activities, the chapter opened by supporting the overall case study context presented in section 7.7.1 with a detailed description of the QUAL team and their project.

Data collection and analysis performed to determine the refined goal rationale was presented in sections 9.3, 9.4 and 9.5. A summary of these findings is presented in table 9-31. Any cells in this table that contain the letter ‘C’ indicate that the particular finding was one of the candidate goal rationale associations proposed in the DScrum model constructed in chapter 5. Similarly, the findings presented in section 9.6 are used to construct a refined value rationale that is presented in table 9-32. Cells with the letter ‘C’ in this table represent goal-value associations that were proposed to be present in the candidate LDScrum model put forward in chapter 5.

The combination of refined goal and value rationales is presented in the next chapter (10). This combination comprises the refined LDScrum model and discussions and conclusions around the research objective are discussed in the context of this model.
<table>
<thead>
<tr>
<th>Core Scrum Practice</th>
<th>Goal</th>
<th>New Product Development</th>
<th>ASD Principles</th>
<th>GSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-1</td>
<td>Scrum Team Configuration</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>CS-2</td>
<td>Initial planning (and ongoing) - create or update product backlog</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-3</td>
<td>Sprint planning - create release backlog</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-4</td>
<td>Sprint planning - create sprint backlog</td>
<td>C</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>CS-5</td>
<td>Sprint planning - define sprint goal</td>
<td>C</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td>C</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-10</td>
<td>Post-Sprint - review meeting</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-11</td>
<td>Post-Sprint - team retrospective</td>
<td>C</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>CS-12</td>
<td>Definition of Done</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9-31: Refined Goal Rationale (Totally Integrated Scrum findings)
<table>
<thead>
<tr>
<th>LDScrum Goal</th>
<th>Code</th>
<th>Description</th>
<th>LDScrum Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>g1</td>
<td>NPD-1</td>
<td>Built-in instability</td>
<td>V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12</td>
</tr>
<tr>
<td>g2</td>
<td>NPD-2</td>
<td>Self-organizing project teams</td>
<td></td>
</tr>
<tr>
<td>g3</td>
<td>NPD-3</td>
<td>Effectively Overlap development phases</td>
<td></td>
</tr>
<tr>
<td>g4</td>
<td>NPD-4</td>
<td>Consistent Multi-learning</td>
<td></td>
</tr>
<tr>
<td>g5</td>
<td>NPD-5</td>
<td>Leverage Subtle control over work tasks</td>
<td></td>
</tr>
<tr>
<td>g6</td>
<td>NPD-6</td>
<td>Effectively disseminate learning throughout Organization</td>
<td></td>
</tr>
<tr>
<td>g7</td>
<td>A-1</td>
<td>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</td>
<td></td>
</tr>
<tr>
<td>g8</td>
<td>A-3</td>
<td>Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.</td>
<td></td>
</tr>
<tr>
<td>g9</td>
<td>A-2</td>
<td>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
<td></td>
</tr>
<tr>
<td>g10</td>
<td>A-4</td>
<td>Business people and developers must work together daily throughout the project.</td>
<td></td>
</tr>
<tr>
<td>g11</td>
<td>A-5</td>
<td>Build projects around motivated individuals. Give them the support they need, and trust them to get the job done.</td>
<td></td>
</tr>
<tr>
<td>g12</td>
<td>A-6</td>
<td>The most efficient and effective method of conveying information within a devt. team is face-to-face conversation.</td>
<td></td>
</tr>
<tr>
<td>g13</td>
<td>A-7</td>
<td>Working software is the primary measure of progress.</td>
<td></td>
</tr>
<tr>
<td>g14</td>
<td>A-8</td>
<td>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</td>
<td></td>
</tr>
<tr>
<td>g15</td>
<td>A-9</td>
<td>Continuous attention to technical excellence and good design enhances agility.</td>
<td></td>
</tr>
<tr>
<td>g16</td>
<td>A-10</td>
<td>Simplicity</td>
<td></td>
</tr>
<tr>
<td>g17</td>
<td>A-11</td>
<td>The best architectures, requirements, and designs emerge from self-organizing teams.</td>
<td></td>
</tr>
<tr>
<td>g18</td>
<td>A-12</td>
<td>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.</td>
<td></td>
</tr>
<tr>
<td>g19</td>
<td>GSD-1</td>
<td>Overcome GSD communication challenges</td>
<td></td>
</tr>
<tr>
<td>g20</td>
<td>GSD-2</td>
<td>Overcome GSD coordination challenges</td>
<td></td>
</tr>
<tr>
<td>g21</td>
<td>GSD-3</td>
<td>Overcome GSD control challenges</td>
<td></td>
</tr>
</tbody>
</table>

Table 9-32: Refined LDScrum value rationale ( Totally Integrated Scrum findings)
Chapter 10 - Conclusions and Recommendations

10.1 Introduction

Figure 10-1 highlights the ‘case report’ WBS segment (2.3 circled below) in the context of the overall project scope (figure 1-7). Although detailed findings for each project in the case are described in WBS segment 2.2 (chapters 8 and 9), the overall case conclusions are reported in this chapter. It commences by using the second level of the overall WBS to summarize the work performed in this research study. This is followed by a presentation and discussion of the case findings and conclusions. This chapter concludes by reflecting upon the research in order to provide both recommendations for future research opportunities and any implications from the findings for industry.

Figure 10-1: Report Empirical Research WBS segment
10.2 Review of research activities

The relationship between LSD and ASD is the research topic investigated in this study. A core reason for this work is to address a gap in the literature that has resulted in much confusion around these two paradigms. The view adopted in this study is that LSD is a broader paradigm than ASD as it considers software development from an overall business perspective. Due to the open nature of the research topic, the high-level objective was refined to an examination of the relationship between LSD and a particular aspect of ASD. This was further specified to stipulate that the study would determine any LSD values that may be present in the performance of Scrum projects by a globally distributed ISD team. The deliverables produced from this work were:

- A research approach to support empirical research into the LSD/ASD relationship.
- A conceptual model linking LSD to this form of ASD (Scrum in GSD).
- A refined model that used empirical findings to relate LSD to this form of ASD.
- A compendium of rich descriptions organized to reflect LSD values that may underpin various goals that motivated Scrum practices.

Figure 10-1 above summarizes the work performed to produce these deliverables. The overall project scope consisted of two main components: ‘1-candidate model construction’ (from literature) and ‘2-refined model reporting’ (from empirical findings). Each of these components were further decomposed into work packages. The activities performed to complete each work package are outlined below. Candidate model construction required four work packages in order to initially determine the parts needed for the model (values, practices, goals) and then to describe how the model was built using associations between these parts:

1.1 An initial set of twelve LSD values was induced from lean principles identified through a genealogical review of literature on the influence of lean thinking in manufacturing, product development, project management and software development. The description of this analysis in chapter 2 includes a presentation of identified lean principles (table 2-1) and the LSD value-set (table 2-3).
1.2 Activities performed to define the Scrum framework and associate it to a base set of goals appropriate to regular and ‘special’ instances of Scrum are presented in chapter 3. A core set of twelve Scrum practices (table 3-3) were identified using selected guides on the application of this method. Eighteen goals were declared using new product development (NPD) characteristics (table 3-1) and the twelve ASD principles associated with the agile manifesto (table 3-2). Various supplementary practices were examined and used to associate the core practices to the eighteen goals (tables 3-10 and 3-11).

1.3 One special instance of Scrum (application in GSD) is explored in chapter 4. An additional three goals were established from GSD literature (table 4-2) and an examination of the use of Scrum in GSD was leveraged to associate the core Scrum practices with these GSD goals (tables 4-9 and 4-10).

1.4 Chapter 5 presents method rationale analysis theory and explains how it was used to construct a conceptual model linking the twelve core Scrum practices to the twenty one goals (Goal rationale). These goals were in turn linked to the LSD values that underpinned them (Value rationale - figures 5-11 and 5-12). This model is called the candidate LDScrum model. It was presented in three method fragments representing pre-sprint (figure 5-13), sprint (figure 5-14) and post-sprint (figure 5-15) Scrum practices.

The second part of this study leveraged the three conceptual method fragments to support empirical research into the relationship between LSD and the application of Scrum in GSD. The scope of this part of the project is comprised of three WBS segments: research approach, implementation and reporting:

2.1 The research strategy described in chapter 6 determined the philosophical assumptions and relevant research methods used to conduct this inquiry. This strategy informed the selection of an interpretive case study approach. The framework used to perform the method is described in chapter 7.
2.2 Implementation of the research involved data collection and analysis of two projects. Rich descriptive findings from each of these projects are presented in chapters 8 and 9.

2.3 This chapter completes the overall study by presenting an integrated view of the findings that have emerged from chapters 8 and 9. A refined version of each of the candidate LDScrum method fragments proposed in chapter 5 (figures 5-13, 5-14 and 5-15) is described. Notable findings that have emerged from the overall study are discussed in the context of how this work has addressed the requirement of the specific research goal.

10.3 Refined LDScrum Model

10.3.1 Presentation concerns

This section presents the refined LDScrum model for the pre-sprint, sprint and post-sprint method fragments. To support comparisons between the candidate and refined model, this section follows a similar structure to the detailed descriptions of the candidate LDScrum model fragments presented in chapter 5. However, the complex relationships uncovered by the empirical findings (chapters 8 and 9) hamper clear reporting of the overall refined LDScrum model. As a result, there are some differences between the presentation of the refined model in this section and the fragment descriptions presented in chapter 5 (sections 5.5.1, 5.5.2 and 5.5.3). Each candidate fragment was described in three ways:

(i) A detailed analysis of the fragment’s practices, goal rationale and value rationale using set theory to describe the associations present in these relationships.

(ii) A diagram illustrating the goal and value rationales - the LDScrum model for the fragment.

(iii) Observations on the value rationale proposed for the fragment.
The refined LDScrum model is described using the latter two components outlined above (diagram and observations). Detailed set descriptions are provided in appendix D.

Furthermore, in chapter 5, the description of the candidate LDScrum model for these three method fragments was prefaced by a rationale for certain tailoring decisions applied to the method rationale analysis framework (section 5.5). A major customization deemed essential to the clarity of the representation of candidate relationships was the removal of the goal achievement hierarchy from each method fragment. This difficulty with the presentation of method rationale is consistent with the concerns expressed by Ågerfalk and Wistrand (2003) in relation to both the effective presentation of method rationale constructs and the complexity in tracking combinations of relationships between primitives. They suggested that representations that use method rationale analysis theory need to be customized as ‘simpler’ frameworks in order to convey information effectively. The candidate LDScrum model presented in chapter 5 is decomposed into three method fragments that each describes a ‘fundamental’ model. The ‘fundamental’ label is applied to these models because they simplify the proposed relationships by avoiding the inclusion of the goal hierarchy in the model. Method fragment decomposition and exclusion of the goal hierarchy are examples of framework tailoring required to leverage this model.

As a result of the relationships identified from the empirical research, it is deemed necessary to further alter the diagram used in the candidate LDScrum model in order to effectively present the findings. The centre of the candidate LDScrum model (figures 5-13, 5-14 and 5-15) presented the proposed associations between core fragment practices and DScrum goals. The left-hand side of the model illustrated the internal value rationale linking internally-oriented goals to internally-oriented values. The right-hand side of the model presented the external value rationale. However, this proposed neat dichotomy has not been realized in the empirical findings. The pursuit of proposed internally-oriented goals has been seen to be underpinned by both internally and externally-oriented LSD values. This mixture of values is also seen in the pursuit of externally-oriented goals. In order to present the refined model in a graphical form that does not become reduced to a scribble of overlapping lines, it is necessary to further tailor this
representation (figures 10-4, 10-7 and 10-10). Both internal and external value rationales are moved to the same side of the diagram. Also, rather than represent the value rationale for all goals, it is deemed sufficient to present only those goals required to display that all LSD values found to underpin goals within that fragment have been linked. For example, in the pre-sprint fragment, it is deemed appropriate to just show the value rationale for g1, g2 and g7 as all LSD values are covered by these three goals. Details of the specific value rationales may be found in tables 8-32 and 9-32. However, the detailed descriptions provided in section 8.6 and 9.6 are the best source of information on how LSD values are interpreted to have underpinned goals.

Finally, the goal rationale of these three method fragments are presented in a tabular format using the same goal-practice structure and notations applied to reporting of the goal rationale of each project in chapters 8 and 9 (tables 8-31 and 9-31). This presents the evolution of the overall I-E-S findings from each individual project investigated and also contrasts the empirical findings with the original proposed associations.

### 10.3.2 Refined LDScrum Pre-Sprint Method Fragment

These results state that all LSD values were found in the combination of empirical findings across the two projects studied in the case. This would indicate that the potential exists for LSD values to be present in pre-sprint activities. However, this very broad observation should be tempered by the recognition that just one empirical finding could result in a value being included in this set. For example, no findings associated with V9 - Data Driven Decisions emerged from the QUAL-NN project. Figure 10-2 shows that pre-sprint planning (CS-4) pursues far more goals than the other practices in this fragment. It also shows that both projects appeared to pursue mainly the same goal-set when performing this practice. Figure 10-4 presents the refined LDScrum model for this method fragment. As stated earlier, for presentation purposes, the value rationale shown is confined to g1, g2 and g7 as this is sufficient to exhibit that all LSD values were found to be present. Both of these figures present high-level broad views of the goal and value rationales for this fragment. However, in order to derive a deeper understanding of the situation, it is necessary to delve into the individual findings. Considerations such as the quantity of findings related to a practice-goal association may add greater meaning to the overall report.
For example, if the evidence indicates that certain goals may be more likely to be pursued than others, then perhaps only the LSD values that underpin those goals should be considered in relation to this fragment? However, this may be too simplistic and lead to the omission of a valid LSD value which is only evident when one considers each thick description in its full context. Specific descriptions and conclusions for each project are reported in detail in chapters 8 and 9.

![Overall Pre-Sprint case goal rationale](image)

**Figure 10-2: Overall Pre-Sprint case goal rationale**

![Comparison of findings counts in Sprint planning (CS-4) practice](image)

**Figure 10-3: Comparison of findings counts in Sprint planning (CS-4) practice**

---

56 The ‘c’ in a cell indicates that the association represented by that cell is proposed in the candidate model.
Figure 10-4: Refined LDScrum for Pre-sprint method fragment

57 To support clarity: (i) Dotted lines indicate associations to GSD specific goals; (ii) CS4 associations indicated using lighter shade; (iii) Only 3 goal-value associations presented to indicate total LSD value coverage (For more detail, cf. appendix D)
10.3.3 Refined LDScrum Sprint Method Fragment

The result of this analysis (figure 10-7) resembles the pre-sprint fragment. All LSD values were found in the combination of empirical findings across the two projects. This indicates that all LSD values may be present in sprint work. Given that CS7 (Sprint - general activities) was found to pursue all but one of the goals then it is little surprise that all values would emerge.

---

<table>
<thead>
<tr>
<th>Core Scrum</th>
<th>Scrum Practices</th>
<th>Challenge</th>
<th>New Product Development</th>
<th>ASD Principles</th>
<th>GSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FIN-NN Project</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td><img src="chart1.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td><img src="chart2.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td><img src="chart3.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td><img src="chart4.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>QUAL-NN Project</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td><img src="chart5.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td><img src="chart6.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td><img src="chart7.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td><img src="chart8.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Overall I-E-S Case Findings</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-6</td>
<td>Sprint - lock features for duration</td>
<td><img src="chart9.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-7</td>
<td>Sprint - general activities</td>
<td><img src="chart10.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-8</td>
<td>Sprint - technical activities</td>
<td><img src="chart11.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-9</td>
<td>Sprint - sprint backlog graph (burndown chart)</td>
<td><img src="chart12.png" alt="Chart" /></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 10-5: Overall Sprint case goal rationale

A specific view of this practice (CS-7) shows that certain goals are reported more than others e.g. 'Overcoming coordination challenges' (G20):

---

Figure 10-6: Contrast of Scrum - general activities (CS-7)

---

58 The ‘c’ in a cell indicates that the association represented by that cell is proposed in the candidate model
Figure 10-7: Refined LDScrum for Sprint method fragment

To support clarity: (i) Dotted lines indicate associations to GSD specific goals; (ii) CS7 associations indicated using lighter shade; (iii) Only 3 goal-value associations presented to indicate total LSD value coverage (For more detail, cf. appendix D)
10.3.4  **Refined LDScrum Post-Sprint Method Fragment**

The post-sprint method fragment is illustrated in figure 10-10. Again, due to the number of practice-goal associations identified, all LSD values are deemed to be present in this fragment. As with the other sections, it is deemed sufficient to present the links between g1, g2 and g7 in order to show total LSD value coverage.

<table>
<thead>
<tr>
<th>Core Scrum</th>
<th>New Product Development</th>
<th>ASD Principles</th>
<th>GSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-1</td>
<td>G-2</td>
<td>G-3</td>
<td>G-4</td>
</tr>
<tr>
<td>G-5</td>
<td>G-6</td>
<td>G-7</td>
<td>G-8</td>
</tr>
<tr>
<td>G-9</td>
<td>G-10</td>
<td>G-11</td>
<td>G-12</td>
</tr>
<tr>
<td>G-13</td>
<td>G-14</td>
<td>G-15</td>
<td>G-16</td>
</tr>
<tr>
<td>G-17</td>
<td>G-18</td>
<td>G-19</td>
<td>G-20</td>
</tr>
<tr>
<td>G-21</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 10-8: Post sprint overall case goal rationale**

This fragment differs from the other two in that all practices of the fragment are well reported in their pursuit of goals. Therefore, figure 10-9 shows both practices. The highest reported goals for each of these practices (G11, G13, G18) are evident in both projects. However, one outlier is that the FIN-NN project reports many instances of the sprint review pursuing coordination complexity whereas this does not arise in QUAL-NN.

**Figure 10-9: Comparison of finding counts for post-sprint fragment practices (CS10 & CS-11)**

The ‘c’ in a cell indicates that the association represented by that cell is proposed in the candidate model.
Figure 10-10: Refined LDScrum model for Post-Sprint method fragment\footnote{To support clarity: (i) Dotted lines indicate associations to GSD specific goals; (ii) CS11 associations indicated using lighter shade; (iii) Only 3 goal-value associations presented to indicate total LSD value coverage (For more detail, cf. appendix D)}
10.4 Insights from research findings

Exploration into the relationship between LSD and ASD was specifically examined in the context of two distributed Scrum projects within one R&D organization. Four specific questions were examined in relation to these two projects. The first three questions investigated whether or not NPD, ASD and GSD goals motivated the performance of Scrum practices. Identification of motivational goals is used to address the fourth inquiry about the presence of LSD values in respondents’ rationale for pursuit of the goals.

The previous section has presented an illustration of the findings with regard to these questions. All LSD values were found to be present in the combined findings of both projects. Furthermore, all LSD values were present in the combined findings of each of the three method fragments.

![Figure 10-11: Breakout of LSD values for each project](image.png)

This overall finding indicates that a relationship between these two paradigms exists but in order to generate useful insights from this work it is necessary to consider the findings at a more granular level. Two LSD values emerged from each team to be the most prevalent among the findings: ‘Effective collaboration’ (V11) and ‘Person focus’ (V4). It appears that the dominant goals anchored by V11 were GSD challenge alleviation goals. Many Scrum practices were
performed to reduce the distance between collaborators. However, other goals were influenced by this value in order to promote teamwork. The empowerment of both individuals and teams was the reason that V4 was noted to underpin the pursuit of many goals. Many of the practices pursued goals that enabled the team to determine their work approach and also promoted continual worker development.

Another LSD value found to be prevalent within both projects is ‘Continuous improvement’ (V5). One would hope for this to be the case given that the review and retrospective Scrum practices facilitate a regular opportunity to propagate information to the wider organization and also to identify areas for improvement within the team. ‘Effective process’ (V10) underpinned many of the QUAL-NN project findings but this value was far less prevalent in the FIN-NN project. The QUAL team have recently adopted Scrum to do their work and are very focussed on process improvements. While V10 was not found to be as influential within the Fin-Limk team, ‘Flow of Value’ (V3) emerged in many of their findings. This may reflect the difference in Scrum maturity between the teams as the FIN-NN project had been using Scrum for much longer and had suffered from earlier situations where too many unfinished stories were present in the last week of the sprint. The Fin-Limk team were less fragmented in their responsibilities and this also could serve to promote awareness of the power of flow as they have more opportunity to visualize an integrated set of stories designed to deliver a piece of value (such as a product feature). Their use of story-pointing and conditions of acceptance techniques demonstrates the increased maturity and unity within this project team. Lack of unity of purpose is an issue understood by the QUAL team and they appear to be using the Scrum framework to decrease the fragmented nature of the team. An additional issue in this regard is that Fin-Limk is a collocated sub-team of a Scrum of Scrums distribution structure whereas the QUAL team structure is more dispersed.

‘Data driven decisions’ (V9) did not really appear to be in the value base of many of the respondents. It did not emerge in the QUAL-NN findings and it was a rather weak value in the few FIN-NN findings where it was considered. This appears to be a potential area for improvement as awareness of this value could lead to effective process modifications. The sprint practice of using a burn-down chart is another area that was identified as a target for additional
training. Both projects used this chart to support their productivity but there appeared to be a lot of confusion around the effectiveness of this device.

One can infer from these results that the use of Scrum within GSD promotes awareness of the LSD values of ‘Effective collaboration’, ‘Person-focus’ and ‘Continuous improvement’. These results show that while these three values are still dominant, awareness of other LSD values is seen in the more experienced Scrum team. Clearly the Fin-Limk team had more experience using Scrum than the QUAL team. This additional maturity supported richer interpretations of the support of Scrum practices for certain LSD values as this team had modified their practices as a result of lessons absorbed during many retrospectives. They also benefitted from engagement with the other teams in the FIN-NN project as they observed failures and successes in Scrum application. Externally-oriented values such as ‘Customer value’, ‘Product excellence’ and ‘Business environment awareness’ emerged from this team. This may reflect their increased use of Scrum. The primary concern of the QUAL team appears to be related to the value of ‘Effective process’ and this would seem to reflect the fact that they are at an early stage in using the Scrum framework.

### 10.5 Recommendations

As stated above, this study produced a research framework and a set of rich descriptions relating LSD to one particular form of ASD: Scrum performed in GSD. This section presents recommendations on how this inquiry may be leveraged to further explore this specific case and also other aspects of the relationship between LSD and ASD.

#### 10.5.1 Deeper investigation of the I-E-S case

As seen from sections 10.3 and 10.4, this study reports on the practices, motivations and values at work in two I-E-S projects using the method fragment as a lens to capture the relationships. A high-level finding from reported motivations has been the dominance of four practices: ‘sprint-
planning’ (CS-4), ‘sprint - general activities’ (CS-7), ‘review’ (CS-10) and ‘retrospective’ (CS-11).

A more detailed analysis of one or more of these practices may reveal further insights into the values that underpin the goals pursued. For example, data has been collected using units of analysis such as team member names, roles and office locations. The ‘method-in-action’ concept described in chapter 5 (section 5.2.2), explained that developer experience influences the manner in which a method is applied. Additional detailed analysis could explore the presence of LSD values in specific practices based upon a developer’s role and experience. This could both extend existing method-in-action research and support a specific thread of LSD/ASD research.

10.5.2 Leverage research framework components

The research framework used method rationale analysis theory to associate a set of LSD values induced from literature to a set of goals. Those goals were associated with a set of core Scrum practices through a hermeneutic process that identified associations with supplementary Scrum GSD practices that were eventually synthesized to the core practice set. It is intended that future research into the high-level objective of resolving the LSD versus ASD debate could leverage the approach and outputs of this study to construct a portfolio of findings relating the broader LSD paradigm to different ASD situations. For example, the LSD value set could be used to support examinations of the relationship between LSD and different ASD methods such as XP or DSDM.

In summary, a number of theoretical constructs have been developed to form the research framework. Each of these theoretical constructs could be used to support further research into both the use of the construct and different topics of investigation:

(i) The proposed LSD value set. This base set was used to anchor a specific strand of research relating lean and agile software development approaches. There is a need for analysis into the relationship between lean and agile practices (Wang, 2011). This research approach could also be used to relate LSD to other ASD methods and enhance
general understanding of the relationship between these two paradigms. At a more general level, research using the LSD value set could be extended to investigate the relationship between LSD and any software development method. From an industrial perspective, the LSD value set may also be applied to software development projects in order to uncover insights into the ‘leanness’ of a particular approach or ‘method-in-action’. This is particularly important in light of Fowler’s (2008) suggestion that developers should focus on lean and agile, not either in isolation.

(ii) **The set of supplementary Scrum GSD practices.** The application of ASD in GSD has been identified as an outstanding research need (Ågerfalk et al., 2009). This practice set was constructed in order to examine the manner in which Scrum could be used to address problems or risks that have been identified with GSD. Supplementary practices were associated with GSD Challenge Alleviation goals. This set could now be used to explore how Scrum might address positive risks or opportunities associated with GSD. Practices could be associated with GSD benefit realization goals.

(iii) **The Method Rationale Analysis Framework.** As stated above, the overall approach of using this framework to construct a model that links LSD values to Scrum goals and practices could be replicated to explore the relationship between LSD and other ASD methods. Further application of this framework would extend the initial work of Ågerfalk and Wistrand (2003) and Ågerfalk and Fitzgerald (2006). This would serve to enlarge the body of knowledge on its application in methodology investigations.

**10.5.3 Further Scrum research**

The rich descriptions generated from the empirical research reflected the interpretations from the members of two distributed project teams using Scrum. The projects selected for this work reflected the broad nature of this inquiry. The teams were configured in two different distributed structures and had considerably different levels of experience and maturity using the Scrum framework.
The relationship between LSD and distributed Scrum could be investigated by replicating this study in more project environments. A programme of research that leverages the LDScrum model to explore Scrum teams that use the distributed Scrum of Scrums structure could enlarge the set of rich descriptions and lead to deeper insights on the specific subject of LSD in relation to this distributed structure. The same applies to the other structure observed in this study: ‘Totally Integrated Scrum’.

The LDScrum approach may be applied to other instances of Scrum in order to reveal insights on these situations. For example, the LDScrum method could be modified to ‘Lean Regulated Scrum’ (LRScrum) in order to investigate the context of Scrum in regulated environments.

An observation from the findings was that the Fin-Limk team had used and customized the application of Scrum much more than the QUAL team. Future studies could replicate the approach linking LSD values to goals and practices of Scrum in order to establish different levels of ‘Scrum application maturity’. One potential outcome of such work could be the proposal of a Scrum maturity model based upon the strength of the relationship between LSD and different Scrum applications.

10.6 Summary

Chapter 10 reviewed the primary objective of this study and the specific lines of inquiry pursued to meet the objective. The structure of all research work undertaken was outlined and the refined LDScrum model was presented. This model illustrated the combined findings from the case. The most prevalent values were highlighted and observations were made on why one project appeared to reflect a more diverse LSD value base than the other. Finally, a series of recommendations was presented on how future research might leverage the research framework and results produced in ‘this’ study.
References


Ambler, S. (2007) ‘Survey says...Agile has Crossed the Chasm.’, Dr. Dobbs Journal, 32 (8), 59.


UL (2005) Lean Thinking/Tools 1 (Course guide in MSc Lean Sigma Systems), University of Limerick, Ireland, ULearning, Enterprise Research Centre.


Appendix A - LSD Value Induction Process
Hermeneutic analysis

Stage 1 - Literature review to identify sources that discuss subjects associated with lean software development. Categorize into themes. Further explanation accompanies the worksheet excerpt supplied below.

Stage 2 - Analyse selected sources to establish core principles suggested by authors. Categorize principles under themes. The conclusion of this stage leads to table 2-1 (chapter 2). Further details and worksheet excerpt supplied below.

Stage 3 - Review the principles worksheet and propose the main and any secondary high-level values that appear to be addressed by the principle. Where appropriate add descriptive information including reasons for value selection. Iterating over this sheet numerous times using the MS-Excel filter functionality helps to establish a broad view of the main values identified. This also ensures consistent naming of values. Excerpt supplied below.

Stage 4 - Build a concise value analysis worksheet. This contained two columns - Principle purpose or feature and associated LSD value. This enabled a rapid iterative review of the LSD value landscape. A key feature used in this analysis was the MS-Excel filter. At any time, this drop-down shows all available values. When happy with this overall sheet, I created a worksheet containing the selected value titles and a detailed description of the value built from the descriptions of the principles used to identify the value. Then go back to the stage 3 worksheet, apply the filter for each LSD value and review all the selected items again to ensure they are most suited to the full description and are not more closely aligned with another value. This iterative approach led to the removal of a proposed value ‘Effective project management’ and all of the principle descriptions originally used to propose this value being applied to a higher-order value ‘Effective Process’. Another example of these modifications was when a final iteration over all entries associated with the ‘Effective Process’ value showed that a value evident in many of these principles was ‘Flow of value’. Ultimately, the final spreadsheet established from stage 4 is found in chapter 2 - table 2-3.
Stage 1 - Worksheet 1: I came up with all these references through trawling books, webs, and conference proceedings. When completed I selected 10 particular sources and looked to see if they had any reference over and above my list that was relevant. If so, then I explored the backward and forward citations of that reference. I identified 6 main themes across the different sources:

- OM  Operations management
- PD  Product development
- PM  Project management
- PI  Process Improvement
- LT  Lean thinking
- Method  Software approach

Much of the summarization of papers/books was done using pen and paper to fill out a dedicated literature reference pad rather than softcopy. It was deemed more effective to use this manual approach rather than typing to immerse myself in the literature.

**Excerpt of Worksheet 1:**

<table>
<thead>
<tr>
<th>Prime Author</th>
<th>Year</th>
<th>Title</th>
<th>Theme</th>
<th>Key points</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPEAR</td>
<td>1999</td>
<td>Decoding the DNA of the TPS</td>
<td>LT</td>
<td>Many companies review TPS - can't replicate system -why? Not culture. Its strict recording of all techniques leads to flexibility (paradox). Community of scientists - scientific method engrained =&gt; willingness to experiment - leads to learning org. All work specified / all cust-suppl interaction direct / pathway for every product simple and direct / all improvement driven by scientific method and guided by teacher and at lowest possible level. How workers learn is by doing - Socratic questioning to gain implicit understanding of work (why it's so hard to transfer lessons).</td>
</tr>
<tr>
<td>CHARETTE, R</td>
<td>2003</td>
<td>Challenging the Fundamental Notions of Software Development</td>
<td>LT</td>
<td>Strategic management definitely</td>
</tr>
<tr>
<td>Prime Author</td>
<td>Year</td>
<td>Title</td>
<td>Theme</td>
<td>Key points</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td>-------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SUTHERLAND, J</td>
<td>2008</td>
<td>The First Scrum: Was it Scrum or Lean? [online]</td>
<td>LT</td>
<td>Scrum came from complex adaptive systems. Initial use was Scrum, xp and hyperproductive teams. Origins (oopsla, org. Patterns). Came from Takeuchi &amp; Nonaka - no mention of Lean in that paper. Punctuated equilibrium - mutations into packages. Scrum &amp; Lean both from CAS. Scrum mechanism to implement lean in CAS that is software.</td>
</tr>
<tr>
<td>HOU, A</td>
<td>1995</td>
<td>Toward Lean Hardware/Software System Development: An Evaluation of Selected Complex Electronic System Development Methodologies</td>
<td>Method</td>
<td>5 lean development methods evaluated. RAPIDS (Spiral type approach); Developer tailoring - incremental; Hardware/software codesign; Cleanroom engineering met all process. Recommended combination of cleanroom and one other (to get design issues)</td>
</tr>
<tr>
<td>HIBBS, C</td>
<td>2009</td>
<td>The art of lean software development: A practical and incremental approach</td>
<td>Method</td>
<td>Overview - Kanban using Scrum</td>
</tr>
<tr>
<td>OHNO, T</td>
<td>1988</td>
<td>The Toyota Production System: Beyond Large-Scale Production</td>
<td>OM</td>
<td>Different principles - core concepts</td>
</tr>
<tr>
<td>REINERTSEN, D</td>
<td>2005</td>
<td>Making R&amp;D Lean</td>
<td>PD</td>
<td>Differences between manufacturing and R&amp;D. Various lean principles and how they may be applied in R&amp;D</td>
</tr>
<tr>
<td>Liker, J</td>
<td>2006</td>
<td>The Toyota Way in Services: The Case of Lean Product Development</td>
<td>PD</td>
<td>Lean gone past manuf. Into other domains; Better lessons for services got from Toyota development system.</td>
</tr>
<tr>
<td>HOLWEG, M</td>
<td>2007</td>
<td>The genealogy of lean production</td>
<td>PD</td>
<td>Good history of TPS; IMVP; Gap between Japan and US in manuf; transplants; Growth in Lean awareness, The “machine”;</td>
</tr>
<tr>
<td>MIDDLETON, P</td>
<td>1995</td>
<td>Maintenance management: From product to process</td>
<td>PI</td>
<td>Two teams A &amp; B -&gt; waste is delay</td>
</tr>
<tr>
<td>MIDDLETON, P</td>
<td>2005</td>
<td>Lean Software Management Case Study: Timberline Inc</td>
<td>PI</td>
<td>Timberline - first in-depth case study</td>
</tr>
<tr>
<td>MIDDLETON, P</td>
<td>2011</td>
<td>Lean Software Management: BBC Worldwide Case Study</td>
<td>PI</td>
<td>BBC - lots of graphs</td>
</tr>
<tr>
<td>HIGHSIMTH, J</td>
<td>2002</td>
<td>Agile software development ecosystems</td>
<td>PM</td>
<td>Interview with Charette &amp; Lean Development overview</td>
</tr>
</tbody>
</table>
**Stage 2:** Excerpt of worksheet. The first iteration of this involved further decomposition of themes to ‘operations mgmt’, ‘product development’, ‘project management’ and ‘software development’. A worksheet listed out a theme and then each author within that theme and finally all the principles identified for each author, including additional information about the principle. This additional information was useful for LSD values induction. The worksheet excerpt shown here reflects one final alteration to remove the author row and add a principle identifier column which indicated the author’s name.

<table>
<thead>
<tr>
<th>Mapping</th>
<th>Purpose</th>
<th>Description - extra information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operations Management</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O1</td>
<td>Continual emphasis on the reduction of waste</td>
<td>7 Wastes: Transportation Inventory Motion Waiting Overproduction Over-processing Defects</td>
</tr>
<tr>
<td>O2</td>
<td>Flow process (Just-in-time)</td>
<td>Ohno addressed manufacturing issues, and this deals with the value of efficiency in that discipline.</td>
</tr>
<tr>
<td>O3</td>
<td>Worker’s stop the line (Autonomation) - built in quality (jidoka and bakka-yoke)</td>
<td></td>
</tr>
<tr>
<td>O4</td>
<td>Respect for people</td>
<td>Managers empower individual skills and teamwork</td>
</tr>
<tr>
<td>O5</td>
<td>Take care in slow growth environment</td>
<td>High-growth promotes economies of scale thinking which may be wasteful or unsuitable in future situations</td>
</tr>
<tr>
<td>O6</td>
<td>Production levelling</td>
<td>Efficiency in manufacturing - relates to project management in LSD.</td>
</tr>
<tr>
<td><strong>Product Development</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>Establish customer-defined value to separate value added from waste</td>
<td>Lean is a never ending journey of waste elimination. Waste is non-value added defined by first defining customer value.</td>
</tr>
<tr>
<td>L5</td>
<td>Develop a “Chief Engineer System” to Integrate Development from start to finish</td>
<td>The chief engineer is the master architect with final authority and responsibility for the entire product development process. The chief engineer is the overarching source of product and process integration.</td>
</tr>
<tr>
<td>L6</td>
<td>6. Organize to balance Functional Expertise and Cross-functional Integration.</td>
<td>Deep functional expertise combined with superordinate goals and the chief engineer system provides the balance sought by matrix organization.</td>
</tr>
<tr>
<td>L7</td>
<td>Develop Towering Technical Competence in all Engineers.</td>
<td>Engineers must have deep specialized knowledge of the product and process that comes from direct experience at the gemba.</td>
</tr>
<tr>
<td>L13</td>
<td>Use powerful tools for standardization and organizational learning</td>
<td>Powerful tools can be simple. Their power comes from enabling standardization which is necessary for organizational learning</td>
</tr>
<tr>
<td>Mapping</td>
<td>Purpose</td>
<td>Description - extra information</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td><strong>Project Management</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>Satisfying the customer is the highest priority</td>
<td>Customer expectations and preferences must be met</td>
</tr>
<tr>
<td>C2</td>
<td>Always provide the best value for money</td>
<td>Define the value - how does software eliminate a risk, overcome a problem or provide a new opportunity</td>
</tr>
<tr>
<td>C3</td>
<td>Success depends on active customer participation</td>
<td>Get buy-in - enable customer to create real-time changes and understand changing condition</td>
</tr>
<tr>
<td>C4</td>
<td>Every lean development project is a team effort</td>
<td>Egoless - multidisciplinary collaboration essential</td>
</tr>
<tr>
<td>C5</td>
<td>Everything is changeable</td>
<td>Decisions on requirements should be as late as possible to provide greatest customer value through ability to deal with change</td>
</tr>
<tr>
<td>C6</td>
<td>Domain solutions, not point solutions</td>
<td>Try to provide solution across markets rather than one specific big solution for one market</td>
</tr>
<tr>
<td>C7</td>
<td>Complete, don’t construct</td>
<td>Assemble - not rewrite (Avoid duplicate work)</td>
</tr>
<tr>
<td>C13</td>
<td>Worker development is vital to lean adoption</td>
<td>This is a form of continual improvement</td>
</tr>
<tr>
<td><strong>Software Development</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF1</td>
<td>(small iterations of fully tested functionality); continuous-flow processing</td>
<td></td>
</tr>
<tr>
<td>MF2</td>
<td>(intense emphasis on requirements elicitation); customer-defined value</td>
<td></td>
</tr>
<tr>
<td>MF3</td>
<td>(estimation and chunking of requirements using function point analysis to support resource allocation to work); design structure matrix to enable flow</td>
<td></td>
</tr>
<tr>
<td>MF4</td>
<td>(Grouping of requirements into “units of work” so that a takt time could be calculated and used to assess team productivity rate); Takt time</td>
<td></td>
</tr>
<tr>
<td>MF5</td>
<td>(reduce distance between collaborators and related processes); Linked processes</td>
<td></td>
</tr>
<tr>
<td>MF6</td>
<td>(consistency of roles, names, work practices etc. to facilitate resource allocation between projects); Standardised procedures</td>
<td></td>
</tr>
<tr>
<td>MF7</td>
<td>(defect prevention using root-cause analysis; scope management enhanced by increased focus on customer needs and context); Eliminate rework</td>
<td></td>
</tr>
<tr>
<td>MF8</td>
<td>(feedback on productivity and errors thus promoting continual learning); Posting results</td>
<td></td>
</tr>
<tr>
<td>MF9</td>
<td>(impartial data collection thus avoiding delays due to meetings and disagreements); Data driven decisions</td>
<td></td>
</tr>
<tr>
<td>MF10</td>
<td>(decompose product components into stories of 3-5 features of 3-5 units of work to be tackled separately by multi-disciplinary teams (developers/QA/Marketing)); Minimize inventory</td>
<td></td>
</tr>
</tbody>
</table>
**Stage 3**: Excerpt of values induction worksheet (note old value “Early delivery of value” that was removed during analysis):

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Purpose</th>
<th>Value</th>
<th>Other values(s)</th>
<th>Description</th>
<th>Reason I selected value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H6</td>
<td>Rapid Development Cycle</td>
<td>Rapid Development Cycle: Release state-of-the-art features to production.</td>
<td>Customer value</td>
<td></td>
<td>Was about early delivery of value - but ultimately this leads to customer value</td>
</tr>
<tr>
<td>H10</td>
<td>Manufacturability</td>
<td>Manufacturability: Ensure that finished product is efficiently producable for markets/customers. (In software applications, deployment)</td>
<td>Customer value</td>
<td></td>
<td>Conscious of how system is to be processed by company. From a deployment/installation perspective, this is of value to customer (especially perceived value (first impressions.....!))</td>
</tr>
<tr>
<td>H11</td>
<td>Supportability</td>
<td>Supportability: Ensure that system can be maintained</td>
<td>Customer value</td>
<td>Impact on overall organization performance</td>
<td>If maintenance difficult, there will be poor customer response to corrective (defects) and perfective (new features) maintenance requests.</td>
</tr>
<tr>
<td>Middleton (2005)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF1</td>
<td>(small iterations of fully tested functionality);</td>
<td>Customer value</td>
<td></td>
<td>continuous-flow processing</td>
<td>Was about early delivery of value - but ultimately this leads to customer value</td>
</tr>
<tr>
<td>MF3</td>
<td>estimation and chunking of requirements</td>
<td>Effective process</td>
<td>Early delivery of value</td>
<td></td>
<td>enable flow</td>
</tr>
<tr>
<td>Criteria</td>
<td>Purpose</td>
<td>Value</td>
<td>Other values(s)</td>
<td>Description</td>
<td>Reason I selected value</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>-------</td>
<td>----------------</td>
<td>-------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>MF4</td>
<td>(Grouping of requirements)</td>
<td>Effective process</td>
<td>Continuous Improvement</td>
<td>Takt time</td>
<td></td>
</tr>
<tr>
<td>MF6</td>
<td>(consistency of roles,</td>
<td>Effective process</td>
<td>Continuous Improvement</td>
<td>Standardised procedures</td>
<td></td>
</tr>
<tr>
<td>MF10</td>
<td>(decompose product components into stories of 3-5 features of 3-5 units)</td>
<td>Customer value</td>
<td>Effective process</td>
<td>Minimize inventory</td>
<td>Was about early delivery of value - but ultimately this leads to customer value</td>
</tr>
<tr>
<td>Middleton (2011) Used a subset of Liker's</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MJ2</td>
<td>Pull work only when needed</td>
<td>Flow of value</td>
<td>Reduce complexity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MJ4</td>
<td>Stop to fix problems</td>
<td>Remove waste</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ohno (1988)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O3</td>
<td>Worker's stop the line (Autonomation) - built in quality</td>
<td>Remove waste</td>
<td></td>
<td></td>
<td>Was quality - now removal of waste due to re-occurrence</td>
</tr>
<tr>
<td>O7</td>
<td>Give machine intelligence (autonomation)</td>
<td>Effective use of technology</td>
<td>Remove waste</td>
<td>Abnormalities cause automatic stoppage</td>
<td></td>
</tr>
<tr>
<td>Charette (2003)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C7</td>
<td>Complete, don't construct</td>
<td>Remove waste</td>
<td>Improvement through reusability</td>
<td>Assemble - not rewrite</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B - Case Study Protocol
CASE STUDY PROTOCOL

Table of Contents

1. OVERVIEW ........................................................................................................................ XI
2. DATA COLLECTION APPROACH ...................................................................................... XI
   DATA COLLECTION ACTIVITIES .......................................................................................... XI
   PREREQUISITES TO INTERVIEW: .................................................................................... XIII
   INTERVIEW FORMAT ......................................................................................................... XIII
3. ANALYSIS APPROACH .................................................................................................... XIV
   INITIAL CODING ............................................................................................................. XIV
   APPROACH GUIDELINES ................................................................................................. XV
4. CODING ........................................................................................................................ XVI
   CONTACT SUMMARY SHEET THEMES ............................................................................ XVI
   TRANSCRIPT CODES ....................................................................................................... XVI
   OBSERVATIONS IN RELATION TO CONSISTENCY OF CERTAIN CODES ..................... XVII
   PLANNED UNITS OF ANALYSIS .................................................................................... XVIII
   HYPOTHESES ................................................................................................................ XVIII
5. DATABASE STRUCTURE .................................................................................................. XIX
6. REFLECTION ON RESEARCH ACTIVITIES .................................................................... XX
7. INTERVIEW BRIEFING LETTER ...................................................................................... XXI
1. **Overview**

This study is a single case study. It reports on a situation where an R&D organization that uses globally distributed teams to develop commercial off-the-shelf software products is implementing an agile method (Scrum) to guide their work activities.

The main unit of analysis is the software development project. For each unit of analysis, three additional sub-units are examined:

(i) The software development team. Each software development project is performed by globally distributed teams. There are two team configurations analysed:
   i. "Scrum of Scrums" (the project team is decomposed into collocated semi-independent sub-teams that perform their own sprints to achieve their goals within the overall project)
   ii. "Totally Integrated Scrum" (The project team contains remote team members who collaborate in the same sprint)

(ii) The software developer. Although the Scrum framework recommends the avoidance of specialized roles and proposes that all developers should be regarded as interchangeable, in reality, developers have a primary skill or role. This role may be as a quality assurance specialist (QA), business analyst (BA) or a software engineer (SE).

(iii) The "method fragment" or sets of Scrum practices performed. It is proposed from theory that Scrum consists of 12 core practices and these have been synthesized into four sets: ‘general considerations’, ‘pre-sprint planning’, ‘sprint’ and ‘post-sprint’. (Three of these are reported although ‘general considerations’ must be analysed to further identify any LSD values that underpin Scrum goals).

2. **Data Collection Approach**

There is only one researcher used in this study. Data is collected and analysed from two primary sources: (i) semi-structured interviews with R&D management and Scrum team members and (ii) relevant project documentation from the development database. Data collection and analysis is conducted in an iterative evolutionary fashion.

**Data collection activities**

1. Organizational interviews held with management to increase understanding of the case, and plan data collection activities. This included the identification of the optimal candidate units of analysis (software projects) and coordinate interviews with the managers of those projects.
Note Initial interview was devised to explore 21 Scrum goals and the presence of 10 lean software development values.

2. Pilot interview held to test interview protocol.

   **Interview Protocol update:** Outcome of this 2 hour interview was that the exploration of lean software development needed to be done separately. Constraint was time/interviewee fatigue.

**SCRUM PROJECT 1: Scrum of Scrums configuration**

3. Project manager interview of first Scrum project conducted to establish a clear understanding of the project and support effective data collection.

   **Decision:** Restrict data collection to one particular sub-team: the Limerick team

4. First round of developer interviews. These interviews were conducted in a semi-structured fashion. They were driven by a theoretical framework. The core aim of each interview:

   4.1 Introduce developer to overall research describing motivations of Scrum and its use in GSD and high-level reference to lean software development values.

   4.2 Ask developer to describe their project and how they perform pre-sprint, sprint and post-sprint activities. Ask them to recall situations where activities went well and when they did not go well. Drilled into questions specific to goals related to Scrum development in GSD.

   **Observation on interview:** This interview did not ask all of the questions on the interview protocol. The questions on the protocol required specific inquiries on 6 NPD goals, 12 core Scrum goals and 3 GSD goals, for each of the three method fragments: pre-sprint, sprint and post-sprint. It was clear that this was far too much for a 90 minute interview. Got general description of activities and specific answers to GSD related inquiries.

   4.3 Conducted two more 90 minute interviews with developers using revised approach. Opened with general request on how they perform each of the three phases. Prompt them about incidents where phases may or may not have gone well. Where opportune, will ask about 6 NPD goals, 12 core Scrum goals or 3 GSD Scrum goals. If any missed, go through these in general at the end of the interview. (Have left the lean values out of the interview other than to just mention them for context).

   **The primary objective is that the interviewee gets an opportunity to describe their observations in relation to the three method fragments of Scrum and that they are asked about how these relate to the 3 GSD**
objectives and that there is at least a general wrap-up enquiring about the
12 core Scrum objectives and 6 NPD goals.

5. Transcribe and analyse interviews (see section 3).
6. Produce initial first version of case study report.
7. Conduct all remaining interviews.
8. Transcribe and analyse interviews.
9. Produce version 2 of case study report
10. Analyse data to apply LSD values
11. Produce final version of case study report for this unit (project)

SCRUM PROJECT 2: Fully Integrated Scrum configuration

12. Project manager interview of second Scrum project conducted to establish a
clear understanding of the project and support effective data collection.
13. Conduct interviews of Limerick-based employees.
14. Transcribe and analyse interviews (see section 3).
16. Transcribe and analyse interviews (see section 3).
17. Produce version 1 of case study report.
18. Apply LSD values
19. Produce final version of case study report for this unit (project)
20. Produce overall case study report merging both projects

Prerequisites to interview:

• Permission for data collection will be acquired from the company under study.
• Each interview will be scheduled in advance.
• Permission to audio-tape each interview will have been pre-approved prior to
  interview.
• Each interviewee will have been briefed in advance about the nature of the
  subject and the content of the interview. One motivation for this is that they have
  a clear idea of which critical incident they wish to discuss. This was a formal
  written invitation with interview details attached.
• Use set of questions as guidelines for interview. They are really questions for the
  researcher.
• Ensure you get permission to conduct follow-up interview, if necessary.

Interview Format

These questions are guidelines for the researcher. Drill-down questions should be
introduced where relevant. Note, as stated above, the interview question-set has
evolved. There are three main sections to the interview process
Section 1 & 2 - This is an interview for managers/supervisors/project managers. It is to get background information around a team and the main projects that they are working on. This may provide insights for the research at hand and also enable more effective data collection from the teams.

Section 3 - This enquires about Scrum goals. It has three subsections that investigate Scrum goals related to new product development characteristics (NPD), goals related to agile software development principles (ASD) and goals related to the use of Scrum to overcome global software development challenges (GSD). These questions are prefixed by the goal number (g1 - g21) and they may be asked in relation to any of the three phases (pre-sprint/sprint/post-sprint).

Section 4 - This investigates the presence of lean software development (LSD) values as drivers for the application of different Scrum practices. These questions are prefixed by the lean value code (L1 - L10).

Details of the manager interview (sections 1 and 2) and the developer interviews (section 3 and 4) are on separate documents.

3. Analysis Approach

Initial coding

- Transcribe interview as quickly as possible
- Create contact summary sheet to capture salient points of the case - assign codes to salient point
- At some point later (one week), manually go through on paper and highlight excerpts and assign codes
- Using software, code raw transcript adding memos to capture any thoughts that emerge during analysis. Two things are achieved with this exercise:
  - Cross checking: There is a form of "cross-checking" going on here - have the contact summary sheet in front of you and the manual printout that you coded. Your code assignments on the contact summary sheet should match the assignments you made on the printout (although the printout may contain additional instances as it is more detailed). The main check is that you did not interpret a particular excerpt differently when analysing it on the second run.
  - More detailed memos: You add detailed memo where relevant to the softcopy
- At this point, each interview is documented by a transcript, contact summary sheet, manual coded transcript and softcopy coded transcript - note that details on coding are outlined in the next section (3.2)
Approach guidelines

- The case findings sheet contains columns that enable data analysis at a very detailed level to reflect the different units of analysis for this case. (See section 3.3 for the planned units of analysis).
- Transfer all coded excerpts to case findings sheet. As part of this process, the analyst adds a memo explaining his interpretation of the code assignment and also what LSD values he feels may underpin the goal. He will also make a judgement on how much he believes the goal was motivated by the practice and whether it was achieved.
  - For each practice, assign "-6", "2", "4" or "6" depending upon the level of influence of the goal on that practice:
    - Contradicts – The practice contradicts the goal
    - Low– The practice was possibly influenced by a goal
    - Medium– It is likely that the practice was performed in order to achieve the goal
    - High– The main reason for performing the practice was to achieve the goal.
  - For each practice, assign "0", "2", "4" or "6" depending upon how well the goal was achieved by performing the practice:
    - Not achieved – The practice did not achieve the goal
    - Low– The practice barely achieved the goal
    - Medium– The practice somewhat achieved the goal
    - High– The goal was clearly observed to have been achieved
- Synthesize case findings for each hypothesis. A separate worksheet is prepared for each of the four hypotheses.
  - Use hypothesis that motivated case to analyse data. Since case is based upon LDScrum framework, use the findings of that framework to identify relevant experiences. Then use judgement to assess findings. A proposed mechanism for assessing data is as follows:
    - For each hypothesis
      i.  Build table of Scrum goals associated with hypothesis. Each row reflects a separate goal.
      ii. Link case findings to table created in (i) above. For each goal, determine the number of findings that were associated with each of the four Scrum method fragments: general, pre-sprint, sprint and post-sprint. These figures will help indicate level of presence of goals across the perceptions of developers.
      iii. Highlight any noted issues with goal motivations or achievement.
      iv. Finally, use personal judgement to interpret findings and be prepared (with explanation) to refute numerical findings if that makes sense. This stage may require a follow-up interview with interviewee.
• Deeper collated analysis of data
  o In Case finding sheet, sort all findings by Goal and then by method fragment.
  o Filter the goal code by each code under analysis.
  o Examine the sheet for that goal and update the hypothesis sheet with the count of findings for the goal that are associated with each method fragment. Note if any findings associated with the method fragment highlighted issues with goal motivation or achievement.
  o Augment with data from team documentation.

• In addressing hypothesis 4 (presence of LSD values):
  o Review memos explaining relationship between method fragment, LDScrum goal and LSD value. Synthesize these memos. Determine further how much of the LSD value is evident in the goal and practice being performed.

• Display data
• Produce traceability matrix from findings to data
• Produce case report
  o Build sub-set of case report for each hypothesis, dividing presentation into hypothesis - method fragment and goals
  o Finally do comparison of LDScrum framework to findings.

4. Coding

  Contact Summary Sheet Themes

Salient points are extracted from each transcript. These are aligned with higher-order themes. These themes reflect the method fragment under discussion and where relevant - if the salient point refers to an NPD, ASD or GSD goal (Note that it could refer to more than one category of goal).

e.g. PreS-NPD

  Transcript codes
All codes should be prefixed by a code that represents the role associated with the data - e.g. individual worker role:

PO/SM/DVBA/DVSE/DVQA/MGR

All codes should be prefixed by a code that represents the phase of the Scrum framework that contains the practice that was associated with the goal

Gen / PreS / S / PostS
So if an interview with a developer from the team whose core skillset is software engineering reveals that the practice of sprint retrospective is motivated by a goal to disseminate learning that would be coded as DVSE-PostS-NPD4

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Description</th>
<th>Code</th>
<th>Associated Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1.1</td>
<td>New Product Development Characteristics Goals</td>
<td>NPD1 - 6</td>
<td>Section 1 - the number corresponds to the goal number: g1 - g6</td>
</tr>
<tr>
<td>H2.1</td>
<td>Agile Software Development Principle Goals</td>
<td>ASD7 - 18</td>
<td>Section 1 - the number corresponds to the goal number: g7 - g18</td>
</tr>
<tr>
<td>H3.1</td>
<td>Global Software Development Challenge Alleviation Goals</td>
<td>GSD19 - 21</td>
<td>Section 1 - the number corresponds to the goal number: g19 - g21</td>
</tr>
<tr>
<td>H4.1</td>
<td>Lean Software Development Values</td>
<td>LV1 - 10</td>
<td>Section 2 - L1 - L10</td>
</tr>
</tbody>
</table>

**Observations in relation to consistency of certain codes**

**Acceptance Criteria meeting**
Respondents describe "acceptance criteria meetings" as a very important practice in their work. Earlier interviews reported this meetings as part of CS4. The Scrum-master interview revealed that it was the desire of the overall project manager that all teams would complete the acceptance criteria at the outset of the sprint. The team deemed the acceptance criteria meeting to be a detailed event akin to a design meeting. When they decided to restrict their work in progress to two user stories, they did not wish to delve into this design meeting at the outset of the sprint. A workaround was devised whereby the team build acceptance criteria at a high-level to address the epic or high level requirement at the sprint outset but they addressed the acceptance criteria for a user-story at the time they began developing that story.

Therefore, this term is used to describe meetings held as the final part of the CS4 sprint planning practice (and as such is coded "PreS") and also is used to describe meetings held during the heart of the sprint and are conducted as design meetings prior to commencing coding on a user story. In these cases, the meetings fall within the CS7 practice (Daily Scrum and other meetings) and as such are coded as "S".

**GSD Challenge Alleviation Goals**
There are 3 GSD challenge alleviation goals addressing challenges related to three software development processes that are very important in effective collaboration:
communication, coordination and control. In order to identify these challenges, they are further decomposed across 3 types of distance, temporal, geographical and socio-cultural. There are different types of challenges that fall within these categories and sometimes a challenge may occur in multiple combinations of process and distance. In the same way, there are comments made by interviewees that could be interpreted as alleviating challenges associated with different distances within a particular process e.g. communication challenges of both a temporal and socio-cultural nature are alleviated by a modular structure. In such instances, the analyst just selects one appropriate code as ultimately alleviation of both temporal and socio-cultural communication challenges are both synthesized to one goal - g19: alleviation of communication challenges.

**Planned Units of Analysis**

**CASE** - R&D division of ERP software product development company
- Interviewed research liaison who is a member of Scrum adoption steering committee and corporate quality manager

**Project 1** - Scrum team developing financials product for ERP add-on.
- Consists of four sub-teams based in four locations. Main team under investigation is the Limerick-based team. Configuration is "Scrum of Scrums".
- Interviewed manager of entire product development group who is also a member of Scrum adoption steering committee

**Project 2** - Quality development group
- Consists of one team with three areas of responsibility. Team is configured in a "Totally Integrated Scrum " structure and employs a ‘management by projects’ approach to their work.
- Interviewed manager of quality team

**Hypotheses**

| RQ1: | How do distributed Scrum projects relate to the characteristics of successful new product development teams as proposed by (Takeuchi and Nonaka, 1986) |
| PR1.1: | **IF** a distributed team performs the ISD method, **SCRUM**, **THEN** they will be seen to exhibit the characteristics of a successful new product development team |
| H1.1: | A distributed team that uses Scrum will exhibit all 6 characteristics of a successful new product development team (Built-in Instability, Self-organization, Subtle control etc.) |

| RQ2: | How do distributed Scrum projects address the principles of Agile software development? |
| PR2.1: | **IF** a distributed team performs the ISD method, **SCRUM**, **THEN** they will |
address the principles of agile software development

<table>
<thead>
<tr>
<th>H2.1: A distributed team that uses Scrum will achieve all 12 goals associated with the principles of ASD</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ3: How do distributed Scrum projects overcome the challenges that face global software development (GSD)</td>
</tr>
<tr>
<td>PR3.1: If a distributed team performs the ISD method, SCRUM, THEN they will overcome many challenges of GSD</td>
</tr>
<tr>
<td>H3.1: A distributed team that uses Scrum will overcome the 3 identified GSD challenges</td>
</tr>
<tr>
<td>RQ4: How do distributed Scrum projects relate to lean software development values?</td>
</tr>
<tr>
<td>PR4.1: If a distributed team performs the ISD method, SCRUM, THEN they will comply with lean software development values</td>
</tr>
<tr>
<td>H4.1: The presence of all 12 identified Lean Software Development (LSD) values will be evident in projects that are performed by distributed teams that use Scrum</td>
</tr>
</tbody>
</table>

5. Database Structure

All data is kept in a Microsoft Windows folder. The folder structure is as follows:

**A - High Level Data Coll and Analysis**
This contains minutes from the initial management meeting to drive case. It also contains the overall data analysis workbook integrating all data analysed across the different units and sub-units. This workbook contains 6 spreadsheets:
(i) Coded excerpts with memos
(ii) Four spreadsheets representing data displays for each hypothesis
(iii) Findings Traceability Matrix

**Primary Unit of Analysis – Project Name**
**A-Data Collection** contains interview protocols
**Context:** any material to set context for project
**Recordings**
**Transcripts**
**B-Data Analysis**
**Contact Summary Sheet.doc:** summary of salient points of interview raw data
**Interview Coding.doc** raw data with codes and memos added as comments during analysis
6. Reflection on research activities

**Data Recording**

Used MS-Windows recorder facility with microphone from laptop for live interviews. Used Skype recorder application for phone interviews.

**Interview Transcription**

Played back recorder mp3 file in slow mode and typed as I listened. I found that an effective foil to typing errors was to go back through quickly and use the MS-Word correction suggestion facility - in many cases it predicted the intended word.

**Transcription analysis - contact summary sheets**

For the most part, I had two screens active, with the original transcript in one screen and the contact summary sheet in the other. I noted each salient point from the transcript. In order to support the identification and recording of themes associated with each salient point, I had a coding guidelines MS Excel workbook. One spreadsheet on this workbook listed explanations for the NPD and ASD goals. Another sheet listed explanations for the GSD goals. By referring to these two sheets, my aim was to interpret the salient points in a consistent fashion and remove unreliability through dependence on memory.
7. Interview Briefing Letter

Hi XXXX,

I have attached a document that I will use to help guide us through the interview next Tuesday. This looks like a huge document and I would say not to worry about it for now. I plan to cover the first 6 pages in our 60-90 minutes. However, I like to keep it all together as it is ultimately the full set of data I wish to analyse. With your permission, I will use my laptop to record the interview and I will transcribe it to MS-Word later. A copy of the transcript is always available should you request it but is confidential to you – I use this raw material to collate with other interviews in order to produce a case study report which ultimately informs the detailed research findings.

The format of the interview is as follows:

I will ask you to consider your current project and describe how you do your job using Scrum. Each iteration of your project can be viewed as the combination of 3 Scrum phases – pre-Sprint planning; Sprint work; post-sprint reflection and review. You will be asked to begin by describing how you do pre-Sprint planning. Then we can look through the attached form to see if you can think of how pre-Sprint planning activities achieved any of the goals listed on the sheet. If you think of any instances where it did, you should try to give an example. We will do the same for Sprint activities (such as daily Scrum) and finally for post-Sprint review activities.

If we get time, I will explore lean software development values in the context of our discussion. However, it is likely we will not get time for this section. I would hope to either do a subsequent interview for this or else perform a group workshop. I will know more when I have analysed the initial data.

Again – I am conscious of your time so don’t feel that you have to review the attached materials in detail prior to the interview. They will merely guide us as we discuss your experiences with Scrum.

Best regards and see you on Tuesday!

Mike
Appendix C - Data Analysis Artefacts
Analysis of core Scrum practice-goal associations

Outline of the analysis process used to develop chapter 3 leading to Scrum practice - goal associations for the regular application of Scrum (tables 3-10 and 3-11). In the interest of space the Worksheets presented below are excerpts.

Stage 1: Establish 12 core Scrum practices from core guides on the Scrum framework

Stage 1.1: Decompose 12 practices into 4 categories - ‘general considerations’; ‘pre-sprint’; ‘sprint’ & ‘post-sprint’.

Stage 2: Establish 21 goals that may be pursued by practices:

Stage 2.1: Determine 6 new product development (NPD) characteristics from HBR paper
Stage 2.2.: Determine 12 agile software development (ASD) principles from Agile manifesto.
Stage 2.3: Determine 3 global software development (GSD) challenges from literature.

Stage 3: Identify supplementary practices that are used or recommended to perform Scrum.

Stage 3.1: Using various authors (books/papers), produce a worksheet listing practices with descriptions and benefits realized by their application (Practices worksheet).
Stage 3.3: Synthesis 2 - looking for deeper explanations - remove duplicates. Reflect 2010 version of Deemer. Also reflecting 2011 version of Schwaber and Sutherland and putting in all the stuff they noted about product backlog. (Practices synthesis 2 worksheet).
Stage 3.4: The NPD characteristics and ASD principles (2.1 & 2.2, above) were inserted (with descriptions) into a ‘Scrum Goals’ worksheet. Practices are grouped into phases. The author was removed and a practice id was added. Finally, practices were associated with principles from the Scrum Goals worksheet. (Practices synthesis 3 worksheet).

Stage 3.5: This synthesis added a core Scrum code in order to reconcile each supplementary practice with the 12 core Scrum practices identified in stage 1.

**Analysis of GSD in Scrum practice - goal associations**

A separate but similar analysis process was performed to associate the 12 core Scrum practices to GSD challenge alleviation goals. Supplementary practices related to the performance of Scrum in GSD were identified from literature. A series of 5 worksheets were used to synthesize these practices and associate them with the GSD goals (table 4-9). Finally, the practices were ‘rolled up’ to core Scrum practices resulting in table 4-10. An excerpt of the final synthesis worksheet is provided below (GSD in Scrum 5 Worksheet).
**Stage 3.1: Practices Worksheet: Initial listing of practices with descriptions (excerpt)**

<table>
<thead>
<tr>
<th>Source</th>
<th>Scrum Practices</th>
<th>Description</th>
<th>Goal/Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highsmith (2002)</td>
<td>Pre-Sprint planning - create product backlog</td>
<td>Create product backlog - list of features needed in product.</td>
<td>Prioritize high-level features to be constructed</td>
</tr>
<tr>
<td></td>
<td>Pre-Sprint planning - create release backlog</td>
<td>Determine list of features to be included in first release of product</td>
<td>Manage stakeholder expectations</td>
</tr>
<tr>
<td></td>
<td>Sprint - lock features for duration</td>
<td>Disallow any external modifications during Sprint. Introduce &quot;stability zone&quot; and promote careful planning.</td>
<td>Reduce time fragmentation. Establish cadence.</td>
</tr>
<tr>
<td>Cohn (2010)</td>
<td>One product owner</td>
<td>Single point of accountability</td>
<td>Rapid decision making</td>
</tr>
<tr>
<td></td>
<td>Small teams</td>
<td>10-12 people (see pg. 179 for references on benefits)</td>
<td>Less coordination needed</td>
</tr>
<tr>
<td></td>
<td>Favour feature teams</td>
<td>Organize teams so that they are fully responsible for a feature</td>
<td>Reduced risk of component integration</td>
</tr>
<tr>
<td>Deemer et al. (2010)</td>
<td>Sprint Burndown chart</td>
<td>Simple visual control of work remaining. Updated every day following re-calibration of remaining time after daily Scrum.</td>
<td>Gauge progress</td>
</tr>
<tr>
<td></td>
<td>Non-extendible Sprint</td>
<td>Never extend - declaration of failure if commitments not met.</td>
<td>Improve planning</td>
</tr>
<tr>
<td></td>
<td>Consistent Sprint Length</td>
<td>Don't change the sprint length.</td>
<td>Regular cadence (heartbeat)</td>
</tr>
<tr>
<td></td>
<td>Retrospective cause analysis</td>
<td>&quot;c&quot; for caused by Scrum; &quot;v&quot; for made visible by Scrum. Help in improving</td>
<td>Continuous improvement</td>
</tr>
</tbody>
</table>
### Stage 3.2 Practices synthesis 1: First synthesis of practices with descriptions (excerpt) (Stage 3.3: Deeper explanations)

<table>
<thead>
<tr>
<th>Source</th>
<th>Scrum Practices</th>
<th>Description</th>
<th>Goal/Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GENERAL CONSIDERATIONS</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cohn (2010)</td>
<td>Single project assignment</td>
<td>Team members exclusively work on one project</td>
<td>Clear prioritization</td>
</tr>
<tr>
<td>Cohn (2010)</td>
<td>Whole-team responsibility</td>
<td>Collective responsibility for success</td>
<td>Increased quality</td>
</tr>
<tr>
<td><strong>INITIAL PLANNING</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sutherland (2006)</td>
<td>Backlog item estimation</td>
<td>To aid prioritization - Use Scrum team - 10 developer days is good size for implementation</td>
<td>Enable effective coordination. Meet customer’s needs.</td>
</tr>
<tr>
<td><strong>PRE-SPRINT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cohn (2010)</td>
<td>DEEP backlog</td>
<td>Detailed appropriate user stories, Estimated, Emergent, Prioritized</td>
<td>Enable planning</td>
</tr>
<tr>
<td><strong>SPRINT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cohn (2010)</td>
<td>Groom the product backlog</td>
<td>10% of each sprint spent reviewing the product backlog (usually by analyst)</td>
<td>Awareness of future change / preparation for next sprint</td>
</tr>
<tr>
<td>Deemer et al. (2010)</td>
<td>Non-extendible Sprint</td>
<td>Never extend - declaration of failure if commitments not met.</td>
<td>Improve planning</td>
</tr>
<tr>
<td><strong>SPRINT - TECHNICAL PRACTICES</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cohen et al. (2004)</td>
<td>*TP - Constant testing</td>
<td>Consistently test product as it is built</td>
<td>Early error detection</td>
</tr>
<tr>
<td><strong>POST-SPRINT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deemer et al. (2010)</td>
<td>Retrospective cause analysis</td>
<td>“c” for caused by Scrum; “v” for made visible by Scrum.</td>
<td>Continuous improvement</td>
</tr>
<tr>
<td><strong>RELEASE</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deemer et al. (2010)</td>
<td>Release Sprint</td>
<td>Clean-up activities for release of product</td>
<td>Effective release management</td>
</tr>
</tbody>
</table>
### Stage 3.4: Practices synthesis 3 Worksheet

<table>
<thead>
<tr>
<th>Code</th>
<th>Scrum Practices</th>
<th>Description</th>
<th>Goal/Benefit</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>GENERAL CONSIDERATIONS</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-5</td>
<td>Single project assignment</td>
<td>Team members exclusively work on one project</td>
<td>Clear prioritization</td>
<td>A-1</td>
</tr>
<tr>
<td>S-6</td>
<td>Whole-team responsibility</td>
<td>Collective responsibility for success</td>
<td>Increased quality</td>
<td>NPD-2; A-9</td>
</tr>
<tr>
<td></td>
<td><strong>INITIAL PLANNING &amp; SPRINT PLANNING</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-16</td>
<td>DEEP backlog</td>
<td>Detailed appropriate user stories, Estimated, Emergent, Prioritized</td>
<td>Enable planning</td>
<td>A-1; A-2</td>
</tr>
<tr>
<td>S-22</td>
<td>Visual progress chart (Scrum board)</td>
<td>4 column wall chart - to-do; in-progress; to-verify; complete. Task migrates on post-it.</td>
<td>Gauge progress</td>
<td>NPD-2; A-8</td>
</tr>
<tr>
<td>S-29</td>
<td>Acceptance test driven development</td>
<td>Using conditions of satisfaction (COS) stipulated by product owner for feature, certain tests created to ensure compliance</td>
<td>Links working model to planned story</td>
<td>A-3; A-10</td>
</tr>
<tr>
<td></td>
<td><strong>SPRINT - TECHNICAL PRACTICES</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-32</td>
<td>Constant testing</td>
<td>Consistently test product as it is built.</td>
<td>Early error detection (reduce technical debt).</td>
<td>A-9</td>
</tr>
<tr>
<td></td>
<td><strong>POST-SPRINT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-42</td>
<td>Retrospective cause analysis</td>
<td>&quot;c&quot; for caused by Scrum; &quot;v&quot; for made visible by Scrum.</td>
<td>Continuous improvement</td>
<td>NPD-4; A-6; A-12</td>
</tr>
<tr>
<td></td>
<td><strong>RELEASE</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-47</td>
<td>Release burndown chart</td>
<td>Analogous to Sprint burndown chart - enables agile release communication</td>
<td>Project management</td>
<td>NPD-2; A-8</td>
</tr>
</tbody>
</table>

The final worksheet in this process (Stage 3.5 Practices synthesis 4) looks similar to above other than the addition of a column representing the core practice associated with each supplementary practice.
**Global Software Development: GSD in Scrum 5 worksheet (excerpt):**

<table>
<thead>
<tr>
<th>ID</th>
<th>Source</th>
<th>Scrum Practices</th>
<th>Description</th>
<th>Ågerfalk Goal</th>
<th>Ågerfalk Goal</th>
<th>Ågerfalk Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>GENERAL CONSIDERATIONS</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSD</td>
<td>Hossain et al. (2011)</td>
<td>Visits - sending &amp; maintaining</td>
<td>For various events/meetings, have distributed team members work face-to-face e.g. One sprint or a sprint-planning day. Useful for both building and maintaining relationships. Variation is to temporarily embed team members in different sites.</td>
<td>com-s1: Avoid cultural misunderstandings that may lead to miscommunication</td>
<td>coord-s1: Establish consistent work practices</td>
<td>con-s1: Avoid morale issues by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>INITIAL PLANNING</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSD</td>
<td>Hossain et al. (2011)</td>
<td>Product backlog</td>
<td>Core practice of <em>Backlog</em> helps overcome lack of critical task awareness.</td>
<td>coord-g1: Promote critical task awareness</td>
<td></td>
<td>con-t1: Single product backlog promotes consistent mechanism across sites for managing high-level scope</td>
</tr>
<tr>
<td></td>
<td>Paasivaara (2009)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSD</td>
<td>Woodward (2010)</td>
<td>Release plan</td>
<td>Despite ideal of release-every-Sprint, many projects will plan release-specific packaging.</td>
<td>coord-g1: Product vision and release plan help distributed team's see &quot;big picture&quot; and relate to it during development</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>PRE-SPRINT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSD</td>
<td>Dodd &amp; Ansari (2010)</td>
<td>“Product Owner Visits“:</td>
<td>Product owner visits to reinforce product vision.</td>
<td>con-g1: Convey clear vision and strategy</td>
<td></td>
<td>con-s1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Source</td>
<td>Scrum Practices</td>
<td>Description</td>
<td>Ågerfalk Goal</td>
<td>Ågerfalk Goal</td>
<td>Ågerfalk Goal</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------</td>
<td>------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------</td>
<td>--------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>GSD</td>
<td>Holmstrom (2006)Passivarra(2009)</td>
<td>Sprint using Daily Scrum</td>
<td>Core practice of Simple planning performed to start Sprint and maintain work with daily Scrum increases &quot;teamness&quot;, reducing geographical coordination distance. (This requires ICT support outlined in practice variations below)</td>
<td>coord-g1: Increase team awareness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSD</td>
<td>Cristal et al. (2008)</td>
<td>Global taskboard</td>
<td>Common tool to plan and coordinate sprint activities (e.g. Wiki or webpage)</td>
<td>coord-t1: Address increased coordination costs</td>
<td>coord-g1: Promote critical task awareness</td>
<td>com-s1: Overcome misunderstandings</td>
</tr>
<tr>
<td>GSD</td>
<td>Woodward (2010)</td>
<td>Product backlog grooming meetings</td>
<td>Weekly meeting - 5-10% of Sprint. For distributed, use balanced team (set of representatives usually. Last week of sprint all distributed team attend)</td>
<td>com-t1: reduced representatives overcome temporal issues</td>
<td>coord-g1: Including everybody at final sprint meeting drives &quot;teamness&quot;</td>
<td></td>
</tr>
<tr>
<td>GSD</td>
<td>Dodda &amp; Ansari (2010)</td>
<td>“Mandatory participation”:</td>
<td>Each site location has to perform mandatory presentation as part of Sprint retrospective</td>
<td>con-s1: Avoid morale issues by managing different perceptions of authority</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSD</td>
<td>Woodward (2010)</td>
<td>Release retrospective</td>
<td>Retrospective spanning many sprints (or entire project)</td>
<td>coord-s1:appreciate practices</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sutherland et al., 2007</td>
<td>Chief Product Owner</td>
<td>Single point of accountability to manage overall product backlog</td>
<td>coord-g1; con-g1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix D - Refined LDScrum Model Details
Pre-sprint refined LDScrum method fragment details.

The three core pre-sprint practices are denoted as: cs3 (create release backlog), cs4 (create sprint backlog) and cs5 (define sprint goal). The empirical findings from the projects described in chapters 8 and 9 are combined to identify the goal rationale for each of these practices and the fragment as a whole. GRcs3 denotes the goal-rationale for ‘create release backlog’: \( \{ (cs3, g_{21}) \} \).

Similarly, GRcs4 = \{ (cs4,g1), (cs4,g2), (cs4,g3), (cs4,g4), (cs4,g5), (cs4,g7), (cs4,g8), (cs4,g11), (cs4,g12), (cs4,g13), (cs4,g14), (cs4,g16), (cs4,g17), (cs4,g19), (cs4,g20), (cs4,g21) \} and GRcs5 = \{ (cs5,g3) , (cs5,g3) \}.

GRSP denotes the goal-rationale for the pre-sprint method fragment. GRSP = GRcs3 \cup GRcs4 \cup GRcs5 = \{ (cs3,g21), (cs4,g1), (cs4,g2), (cs4,g3), (cs4,g4), (cs4,g5), (cs4,g7), (cs4,g8), (cs4,g11), (cs4,g12), (cs4,g13), (cs4,g14), (cs4,g16), (cs4,g17), (cs4,g19), (cs4,g20), (cs4,g21), (cs5,g3) , (cs5,g5) \}.

The range of this set represents the goals that may be achieved by performance of the pre-sprint method fragment: GSP = Ran(GSP) = \{ g1, g2, g3, g4, g5, g7, g8, g11, g12, g13, g14, g16, g17, g19, g20, g21 \}.

Each of the goals present in GSP may be sought by a developer based upon the values held by that developer. The empirical findings widened the set of values that may underpin the pursuit of various goals. Therefore, potential value rationale available to developers engaged in pre-sprint (or sprint planning) activities, VRSP was found to be as follows:

\[ \{ (g1,v1), (g1,v2), (g1,v3), (g1,v10), (g1,v11), (g2,v2), (g2,v3), (g2,v5), (g2,v7), (g2,v10), (g2,v12), (g2,v13), (g2,v4), (g2,v10), (g3,v11), (g4,v4), (g4,v3), (g4,v11), (g5,v1), (g5,v2), (g5,v10), (g5,v11), (g7,v1), (g7,v3), (g7,v5), (g7,v10), (g8,v2), (g8,v3), (g8,v5), (g8,v10), (g9,v10), (g10,v1), (g11,v2), (g11,v12), (g12,v11), (g13,v1), (g13,v2), (g13,v4), (g13,v5), (g13,v10), (g14,v1), (g14,v2), (g14,v3), (g14,v5), (g14,v10), (g15,v11), (g16,v12), (g16,v1), (g16,v2), (g16,v5), (g16,v10), (g16,v11), (g16,v12), (g17,v3), (g17,v5), (g17,v6), (g17,v11), (g17,v12), (g18,v10), (g18,v2), (g18,v10), (g18,v12), (g18,v11), (g19,v10), (g19,v12), (g19,v11), (g19,v12) \} \]

The set of potential values in the pre-sprint value base, VSP is Ran(VSP) = \{ v_{15}, v_{2}, v_{2}, v_{2}, v_{3}, v_{6}, v_{7}, v_{8}, v_{10}, v_{11}, v_{12} \}.

xxx | Page
Sprint refined LDScrum method fragment details.

The sprint method fragment consists of four core practices: CS-6 (Lock sprint features), CS-7 (Sprint - general activities), CS-8 (Sprint - technical activities) and CS-9 (Sprint backlog graph). GR_{cs6} denotes the goal-rationale for ‘lock features’: \{(cs_6,g_4), (cs_6,g_7), (cs_6,g_8), (cs_6,g_14), (cs_6,g_16)\}. The combined findings from both projects only found one goal (g1) to be absent from general sprint activities. Therefore GR_{cs7} = \{(cs_7,g_2), (cs_7,g_3), (cs_7,g_4), (cs_7,g_5), (cs_7,g_6), (cs_7,g_7), (cs_7,g_8), (cs_7,g_9), (cs_7,g_{10}), (cs_7,g_{11}), (cs_7,g_{12}), (cs_7,g_{13}), (cs_7,g_{14}), (cs_7,g_{15}), (cs_7,g_{16}), (cs_7,g_{17}), (cs_7,g_{18}), (cs_7,g_{19}), (cs_7,g_{20}), (cs_7,g_{21})\}. GR_{cs8} presents the goal rationale of sprint-technical = \{(cs_8,g_2), (cs_8,g_3), (cs_8,g_5), (cs_8,g_{10}), (cs_8,g_{11}), (cs_8,g_{12}), (cs_8,g_{13}), (cs_8,g_{14}), (cs_8,g_{16})\}. Finally, GR_{cs9} refers to the sprint backlog graph = \{(cs_9,g_3), (cs_9,g_{11}), (cs_9,g_{12}), (cs_9,g_{13}), (cs_9,g_{14}), (cs_9,g_{20})\}. GR_{S} denotes the goal-rationale for the sprint method fragment: GR_{S} = GR_{cs6} \cup GR_{cs7} \cup GR_{cs8} \cup GR_{cs9} = \{(cs_9,g_3), (cs_9,g_7), (cs_9,g_{12}), (cs_9,g_{13}), (cs_9,g_{14}), (cs_9,g_{20})\}.

The range of this set represents the goals that may be achieved by performance of the sprint method fragment: G_{S} = \text{Ran}(GR_{S}) = \{(g_2,v_2), (g_2,v_3), (g_2,v_4), (g_2,v_5), (g_2,v_9), (g_2,v_{10}), (g_2,v_{12}), (g_3,v_4), (g_3,v_{10}), (g_4,v_4), (g_4,v_5), (g_4,v_{11}), (g_5,v_1), (g_5,v_2), (g_5,v_4), (g_5,v_{10}), (g_6,v_5), (g_6,v_{11}), (g_6,v_{12}), (g_7,v_7), (g_7,v_8), (g_7,v_{10}), (g_8,v_2), (g_8,v_3), (g_8,v_7), (g_8,v_{10}), (g_9,v_1), (g_9,v_2), (g_9,v_3), (g_9,v_7), (g_9,v_8), (g_9,v_{10}), (g_{10},v_1), (g_{10},v_6), (g_{10},v_8), (g_{10},v_{10}), (g_{10},v_{11}), (g_{11},v_2), (g_{11},v_3), (g_{11},v_4), (g_{11},v_{10}), (g_{11},v_{11}), (g_{11},v_{12}), (g_{12},v_{11}), (g_{13},v_1), (g_{13},v_2), (g_{13},v_3), (g_{13},v_4), (g_{13},v_5), (g_{13},v_9), (g_{13},v_{10}), (g_{14},v_1), (g_{14},v_2), (g_{14},v_3), (g_{14},v_4), (g_{14},v_5), (g_{14},v_7), (g_{14},v_9), (g_{14},v_{10}), (g_{14},v_{11}), (g_{14},v_{12}), (g_{15},v_2), (g_{15},v_5), (g_{15},v_{10}), (g_{16},v_1), (g_{16},v_2), (g_{16},v_3), (g_{16},v_8), (g_{16},v_{10}), (g_{16},v_{11}), (g_{16},v_{12}), (g_{17},v_4), (g_{17},v_5), (g_{17},v_6), (g_{17},v_{11}), (g_{18},v_5), (g_{18},v_{10}), (g_{19},v_{11}), (g_{19},v_{12}), (g_{20},v_4), (g_{20},v_{10}), (g_{20},v_{12}), (g_{21},v_1), (g_{21},v_2), (g_{21},v_12)\}.

Developers seek to achieve certain goals as a result of those goals being anchored by particular values held by the developer. Given that only one goal is absent from the goal rationale for this method fragment, it is likely that all LSD values will be associated with the fragment. The value rationale available to developers engaged in the performance of sprint activities, VR_{S} = \{(v_2,v_4), (v_2,v_5), (v_3,v_4), (v_3,v_v), (v_4,v_5), (v_4,v_7), (v_5,v_1), (v_5,v_2), (v_5,v_4), (v_5,v_{10}), (v_6,v_5), (v_6,v_{11}), (v_6,v_{12}), (v_7,v_7), (v_7,v_8), (v_7,v_{10}), (v_8,v_2), (v_8,v_3), (v_8,v_7), (v_8,v_{10}), (v_9,v_1), (v_9,v_2), (v_9,v_3), (v_9,v_7), (v_9,v_8), (v_9,v_{10}), (v_{10},v_1), (v_{10},v_6), (v_{10},v_8), (v_{10},v_{10}), (v_{10},v_{11}), (v_{11},v_2), (v_{11},v_3), (v_{11},v_4), (v_{11},v_{10}), (v_{11},v_{11}), (v_{11},v_{12}), (v_{12},v_{11}), (v_{13},v_1), (v_{13},v_2), (v_{13},v_3), (v_{13},v_4), (v_{13},v_5), (v_{13},v_9), (v_{13},v_{10}), (v_{14},v_1), (v_{14},v_2), (v_{14},v_3), (v_{14},v_4), (v_{14},v_5), (v_{14},v_7), (v_{14},v_9), (v_{14},v_{10}), (v_{14},v_{11}), (v_{14},v_{12}), (v_{15},v_2), (v_{15},v_5), (v_{15},v_{10}), (v_{16},v_1), (v_{16},v_2), (v_{16},v_3), (v_{16},v_8), (v_{16},v_{10}), (v_{16},v_{11}), (v_{16},v_{12}), (v_{17},v_4), (v_{17},v_5), (v_{17},v_6), (v_{17},v_{11}), (v_{18},v_5), (v_{18},v_{10}), (v_{19},v_{11}), (v_{19},v_{12}), (v_{20},v_4), (v_{20},v_{10}), (v_{20},v_{12}), (v_{21},v_1), (v_{21},v_2), (v_{21},v_{12})\}.

The set of potential values in the sprint value base, V_S is Ran(VR_S) = \{v_{1}, v_{2}, v_{3}, v_{4}, v_{5}, v_{6}, v_{7}, v_{8}, v_{9}, v_{10}, v_{11}, v_{12}\}. 

xxxii | Page
Post-sprint refined LDScrum method fragment details.

There are two practices addressed in this method fragment. The goal rationale for CS-10 (Review) is $GR_{cs10} = \{(cs_{10},g_1), (cs_{10},g_4), (cs_{10},g_5), (cs_{10},g_7), (cs_{10},g_9), (cs_{10},g_{11}), (cs_{10},g_{13}), (cs_{10},g_{14}), (cs_{10},g_{16}), (cs_{10},g_{17}), (cs_{10},g_{18}), (cs_{10},g_{20}), (cs_{10},g_{21})\}$.

The goal rationale of the second practice, CS-11 (Retrospective) is $GR_{cs11} = \{(cs_{11},g_2), (cs_{11},g_4), (cs_{11},g_{12}), (cs_{11},g_{14}), (cs_{11},g_{17}), (cs_{11},g_{18}), (cs_{11},g_{20})\}$. $GR_{PS}$ denotes the goal-rationale for the post-sprint method fragment: $GR_{cs10} \cup GR_{cs11} = GR_{PS} = \{(cs_{10},g_1), (cs_{10},g_4), (cs_{10},g_5), (cs_{10},g_6), (cs_{10},g_7), (cs_{10},g_9), (cs_{10},g_{11}), (cs_{10},g_{13}), (cs_{10},g_{14}), (cs_{10},g_{15}), (cs_{10},g_{16}), (cs_{10},g_{17}), (cs_{10},g_{18}), (cs_{10},g_{20})\}$. $G_{PS}$ represents the range of goals that may be achieved by performance of the post-sprint method fragment: $G_{PS} = \text{Ran}(GR_{PS}) = \{g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11}, g_{12}, g_{13}, g_{14}, g_{15}, g_{16}, g_{17}, g_{18}, g_{19}, g_{20}, g_{21}\}$.

The value rationale available to developers engaged in the performance of post-sprint activities, $VR_{PS}$ is as follows:

$\{ (g_1,v_1), (g_1,v_4), (g_1,v_7), (g_1,v_{10}), (g_1,v_{11}), (g_2,v_2), (g_2,v_3), (g_2,v_5), (g_2,v_9), (g_2,v_{10}), (g_2,v_{11}), (g_2,v_{12}), (g_4,v_4), (g_4,v_5), (g_4,v_{11}), (g_5,v_1), (g_5,v_2), (g_5,v_4), (g_5,v_{10}), (g_6,v_5), (g_6,v_{11}), (g_6,v_{12}), (g_7,v_1), (g_7,v_8), (g_7,v_{10}), (g_9,v_1), (g_9,v_2), (g_9,v_3), (g_9,v_7), (g_9,v_8), (g_9,v_{10}), (g_{10},v_1), (g_{10},v_2), (g_{10},v_3), (g_{10},v_4), (g_{10},v_{10}), (g_{11},v_2), (g_{11},v_3), (g_{11},v_4), (g_{11},v_{10}), (g_{11},v_{12}), (g_{12},v_{11}), (g_{13},v_1), (g_{13},v_2), (g_{13},v_3), (g_{13},v_4), (g_{13},v_5), (g_{13},v_9), (g_{13},v_{10}), (g_{14},v_1), (g_{14},v_2), (g_{14},v_3), (g_{14},v_{14}), (g_{14},v_5), (g_{14},v_7), (g_{14},v_9), (g_{14},v_{10}), (g_{14},v_{11}), (g_{14},v_{12}), (g_{15},v_2), (g_{15},v_6), (g_{15},v_{10}), (g_{16},v_1), (g_{16},v_2), (g_{16},v_3), (g_{16},v_8), (g_{16},v_{10}), (g_{16},v_{11}), (g_{16},v_{12}), (g_{17},v_4), (g_{17},v_5), (g_{17},v_6), (g_{17},v_{11}), (g_{18},v_5), (g_{18},v_{10}), (g_{19},v_11), (g_{19},v_{12}), (g_{20},v_4), (g_{20},v_{10}), (g_{20},v_{11}), (g_{20},v_{12}), (g_{21},v_{10}), (g_{21},v_11), (g_{21},v_{12})\}$.

All values are evident in the post-sprint value rationale, $V_{PS}$. This is seen by noting that $\text{Ran}(VR_{PS}) = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_{10}, v_{11}, v_{12}\}$. 
Appendix E - Findings Data Dictionary (Excerpt)
Appendix F - Author Publications related to thesis
<table>
<thead>
<tr>
<th>Subject</th>
<th>Publication</th>
<th>Chapter</th>
</tr>
</thead>
</table>